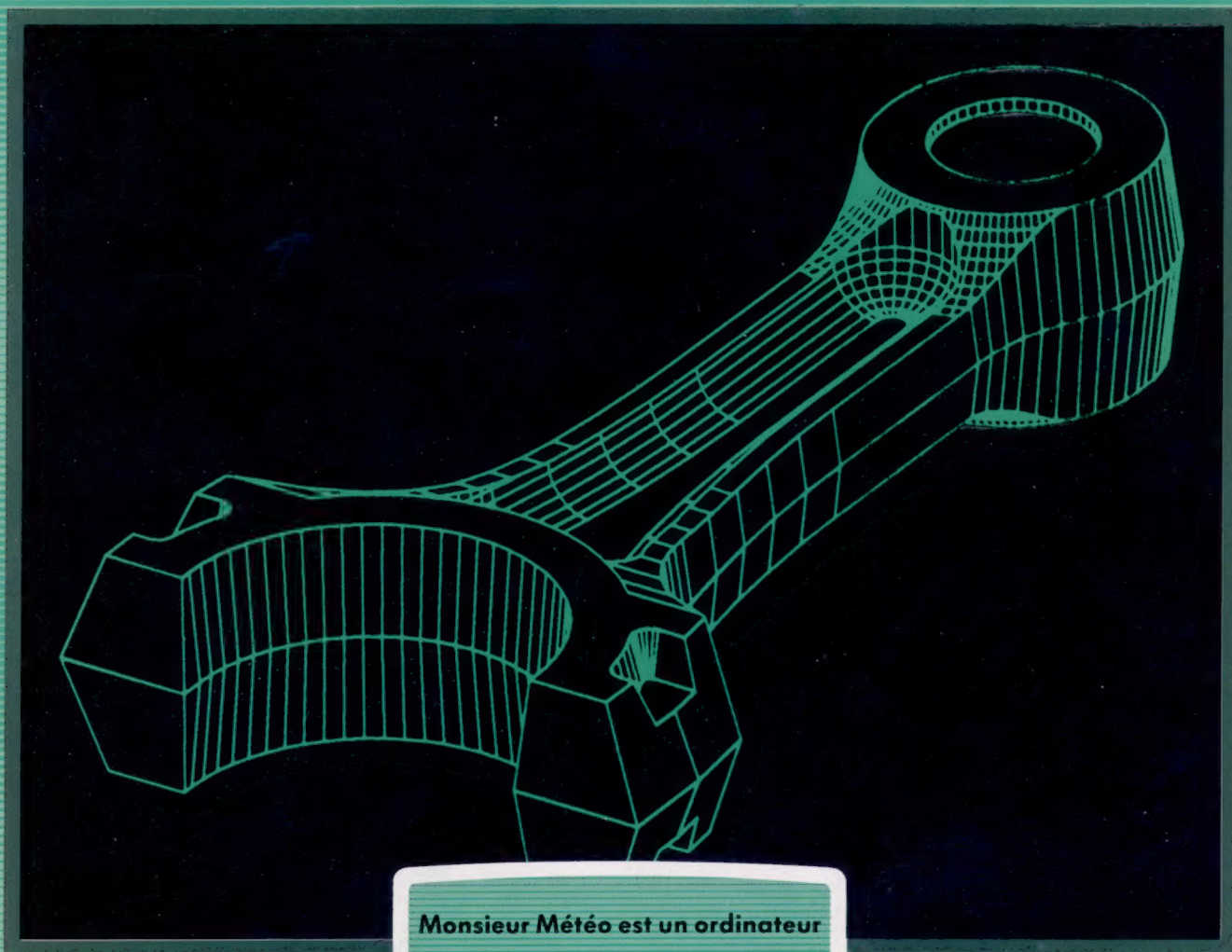


abc

N° 13

COURS
D'INFORMATIQUE
PRATIQUE
ET FAMILIALE

INFORMATIQUE



Monsieur Météo est un ordinateur

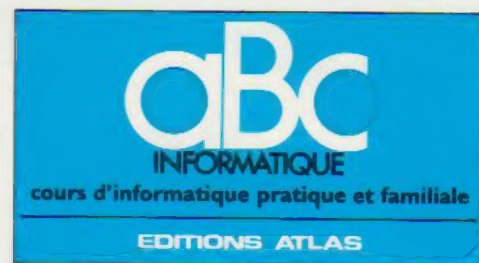
Le Sord M5

Faites votre flipper

Les missiles de croisière

EDITIONS
ATLAS

Dans toutes les librairies



Édité par ÉDITIONS ATLAS s.a., tour Maine-Montparnasse, 33, avenue du Maine, 75755 Paris Cedex 15. Tél. : 538-52-70.

Belgique : ÉDITIONS ATLEN s.a., Bruxelles.

Canada : ÉDITIONS ATLAS CANADA Ltée, Montréal Nord.

Suisse : FINABUCH s.a., ÉDITIONS TRANSALPINES, Mezzovico.

Realisé par EDENA s.a., 29, boulevard Edgar-Quinet, 75014 Paris Tél. : 320-15-01.

Direction éditoriale : J.-Fr. Gautier. Service technique et artistique : F. Givone et J.-Cl. Bernar. Iconographie : J. Pierre. Correction : B. Noël.

VENTE

Les numéros parus peuvent être obtenus chez les marchands de journaux ou, à défaut, chez les éditeurs, au prix en vigueur au moment de la commande. Ils resteront en principe disponibles pendant six mois après la parution du dernier fascicule de la série. (Pour toute commande par lettre, joindre à votre courrier le règlement, majoré de 10 % de frais de port.)

Pour la France, s'adresser à ÉDITIONS ATLAS, tour Maine-Montparnasse, 33, avenue du Maine, 75755 Paris Cedex 15. Tél. : 538-52-70.

Pour les autres pays, s'adresser aux éditeurs indiqués ci-dessous.

SOUSCRIPTION

Les lecteurs désirant souscrire à l'ensemble de cet ouvrage peuvent s'adresser à :

France : DIFFUSION ATLAS, 3, rue de la Taye, 28110 Lucé. Tél. : (37) 35 40 23.

Belgique : ÉDITIONS ATLEN s.a., 55, avenue Huart-Hamoir, 1030 Bruxelles. Tél. : (02) 242-39-00. Banque Bruxelles-Lambert, compte n° 310-0018465-24 Bruxelles.

Canada : ÉDITIONS ATLAS CANADA Ltée, 11450 boulevard Albert-Hudon, Montréal Nord, H 1G 3J9.

Suisse : FINABUCH s.a., ÉDITIONS TRANSALPINES, zona industriale 6849 Mezzovico-Lugano. Tél. : (091) 95-27-44.

RELIEZ VOS FASCICULES

Des reliures mobiles, permettant de relier 12 fascicules, sont en vente en permanence chez votre marchand de journaux.

ATTENTION : ces reliures, présentées sans numérotation, sont valables indifféremment pour tous les volumes de votre collection. Vous les numéroterez vous-même à l'aide du décalque qui est fourni (avec les instructions nécessaires) dans chaque reliure.

En vente tous les vendredis. Volume II, n° 13.

ABC INFORMATIQUE est réalisé avec la collaboration de Tristan Mordret (secrétariat de rédaction), S.I.-André Laroche (traduction), Ghislaine Goullier (fabrication), Marie-Claire Jacquet (iconographie), Patrick Boman (correction), Crédit photographique, couverture : Matra/Data-Vision.

Directeur de la publication : Paul Bernabeu. Imprimé en Italie par I.G.D.A., Officine Grafiche, Novara. Distribution en France : N.M.P.P. Tax. Dépôt légal : mars 1984, 30843. Dépôt légal en Belgique : D/84/2783/27.

© Orbis Publishing Ltd., London.
© Editions Atlas, Paris, 1984.

A NOS LECTEURS

En achetant chaque semaine votre fascicule chez le même marchand de journaux, vous serez certain d'être immédiatement servi, en nous facilitant la précision de la distribution. Nous vous en remercions d'avance.

Les Éditions Atlas

Histoire des BATAILLES NAVALES

de la voile aux missiles

Histoire des batailles navales

de la voile aux missiles

Des duels de l'Antiquité au conflit des Malouines, cet ouvrage retrace vingt-cinq siècles de guerre sur mer. Ce livre, vivant et précis, illustré de documents anciens et récents sur les navires, les hommes et les opérations qui ont marqué le contrôle de la mer, restitue aux conflits maritimes leur place primordiale dans l'histoire de la guerre.

Un volume relié, sous jaquette illustrée. 224 pages. 97 photos, 2 dessins et 2 cartes en couleurs. 110 photos et 4 cartes en noir et blanc.

Format : 22,5 × 29,5 cm.

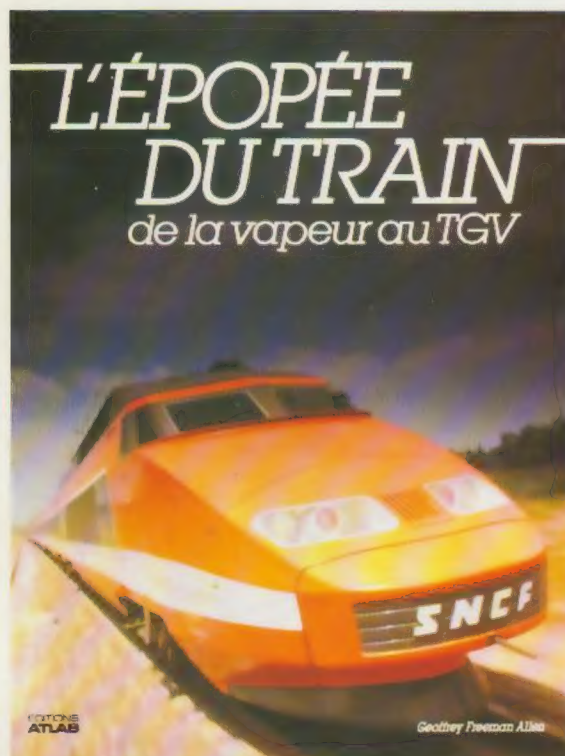
Dans toutes les librairies

L'Épopée du train

De la vapeur au TGV

Depuis la mise en service au tout début du XIX^e siècle des premières locomotives à vapeur jusqu'au succès récent du TGV, les chemins de fer ont voulu répondre au défi économique. A travers les techniques de construction, l'évolution des réseaux et la vie d'hommes volontaires, cet ouvrage retrace la formidable histoire des trains. Il passionnera ceux pour qui les chemins de fer sont synonyme de progrès et d'évasion.

1 volume relié, sous jaquette illustrée. 304 pages. 104 photos en couleurs. 239 photos en noir et blanc. 11 cartes en couleurs. 37 dessins en couleurs.
Format : 22,5 × 29 cm.





As du flipper

Le programme Pinball Construction Set, logiciel remarquablement réussi, vous permet de concevoir et d'utiliser vos propres jeux de flipper sur un ordinateur Apple.

Industrie dynamique, la micro-informatique devrait normalement introduire sans cesse des idées entièrement nouvelles. Pourtant, il est rare de rencontrer un programme qui se démarque des autres de façon sensible à la fois sur le plan de la conception et sur celui de la qualité. Le logiciel Pinball Construction Set (PCS) de Budgéco est un exemple de produit réellement révolutionnaire. Fonctionnant sur Apple II 48 K, avec un lecteur de disquettes et un manche à balai, ce logiciel exécute des fonctions apparemment très simples. Il propose à l'utilisateur l'image d'une table de flipper sans accessoires et un menu de trente-huit dispositifs différents qui permettent de construire un flipper personnalisé. Le programme comporte en plus un menu de fonctions où l'on peut choisir ses outils.

Après avoir rempli la surface de jeu — vous pouvez définir jusqu'à cent vingt-huit positions sur cette surface —, il ne vous reste plus qu'à essayer. Pour ce faire, vous devez utiliser le manche à balai. Le programme peut accepter jusqu'à quatre joueurs, mais chacun ne dispose

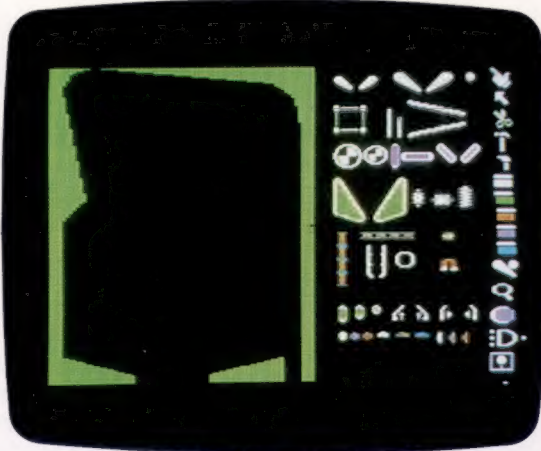
commandées à l'aide du manche à balai. Le premier outil à utiliser est une main. Elle sert à désigner un objet dans le menu « accessoires » (comme un flipper ou une bande) et, quand vous appuyez sur le bouton de la manette, la main saisit l'objet désigné et le tire vers l'endroit désiré sur la table, jusqu'à ce que vous relâchiez le bouton de mise à feu et déterminiez ainsi son emplacement définitif.

L'intérêt de cette démarche tient également au fait que vous ne déplacez pas uniquement les données qui définissent la forme de l'objet, mais aussi l'ensemble des règles qui régissent ses interventions lors du déroulement du jeu. Un flipper, par exemple, se déplace toujours de 45° vers le haut, puis vers le bas. Une bande repousse la balle avec une accélération qui correspond à un facteur de rebondissement définissable. La boule obéit aux lois de la dynamique newtonienne, et retombe vers le bas de la table conformément aux lois de la gravité.

Mais, cela dit, il existe également un outil (représenté judicieusement par une planète par-

Faites-les vous-même

Le programme Pinball Construction Set affiche une table de flipper sans accessoires, divers types de dispositifs (bandes, cibles, flippers et ainsi de suite), et dans la colonne de droite les outils servant à placer les objets sur la table. Cette colonne renferme également certaines fonctions servant à définir la dimension, la forme, la couleur et les diverses interactions des pièces entre elles, ainsi qu'à sauvegarder sur disquettes les jeux. (Cl. Ian McKinnell.)



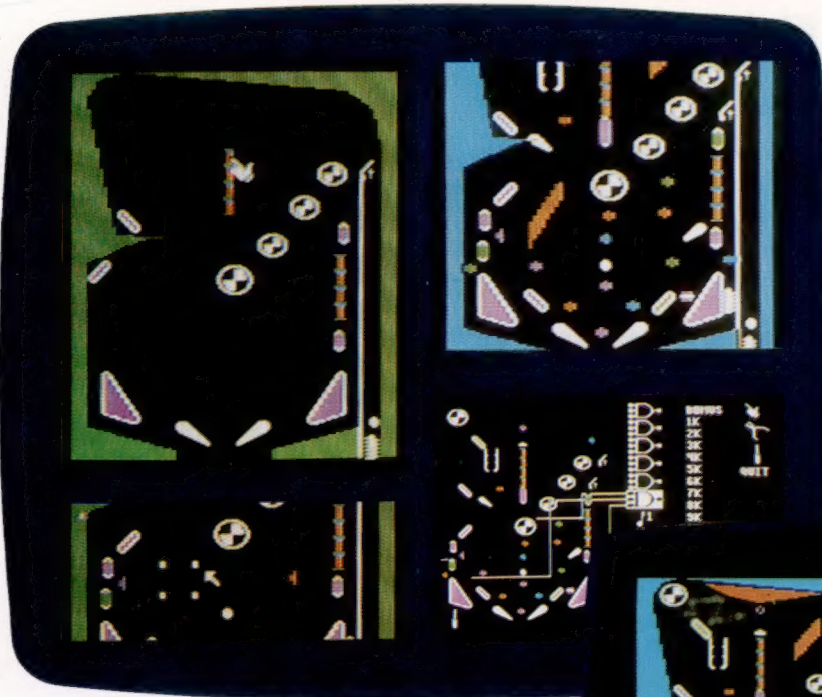
que d'une seule boule, et non de trois comme sur la plupart des flippers. En outre, il n'y a pas de fonction « boule gratuite ». A la fin de la partie, vous pressez sur la touche ESCAPE pour vous faire revenir au menu. Vous pouvez continuellement améliorer la surface de jeu. Ajoutons qu'il n'y a pas de limite au nombre de fois où vous désirez jouer avec un modèle de flipper que vous avez choisi auparavant.

La conception et le mode d'exécution du PCS en font un logiciel vraiment facile à utiliser. Dès le chargement du programme (l'utilisateur n'a qu'à introduire le disque et à appuyer sur RETURN), presque toutes les interventions sont

Jeux d'enfants

PCS vous permet de créer des effets sonores et visuels comparables aux jeux d'arcades. Mais il est en fait plus amusant de concevoir les jeux que de les utiliser... (The Image Maker.)





Étape par étape

Ces quatre images illustrent les diverses étapes de la construction d'un jeu de flipper. Tout d'abord, les pièces élémentaires sont installées, puis un polygone est ajouté pour former un îlot central. Le polygone est déformé et coloré de couleur orange. Finalement, certains objets sont reliés.

Prêt à fonctionner

Dès que le jeu est construit à l'écran, il peut être sauvegardé sur disquette. Le logiciel initial (PCS) n'est pas nécessaire pour exécuter le jeu créé. (Cl. Ian McKinnell.)



tiellement ensoleillée) qui vous permet de définir les paramètres environnementaux du monde réel (force gravitationnelle, par exemple, et même le temps; cette fonction est également contrôlée par la manette). Chaque valeur est définie en modifiant une position sur une échelle, tout comme le volume sonore d'une chaîne-hi-fi peut être réglé à l'aide d'un commutateur à glissement qu'on pousse vers le haut ou vers le bas.

Toutes les fonctions qu'on retrouve généralement dans un programme graphique bien conçu sont aussi disponibles. Vous pouvez utiliser des « outils » qui servent à étirer et à déformer des lignes en les tendant entre deux points déterminés (tension-caoutchouc), à colorer des surfaces avec l'une des couleurs de la palette et à agrandir de petites portions de l'image graphique afin de pouvoir inclure plus de détails.

Mais ce qui distingue surtout le programme Pinball Construction Set, en plus de ses diverses fonctions et de son potentiel, c'est sa philosophie d'utilisation. Cette programmation orientée vers les objets — où chaque élément du logiciel comporte les détails de son fonctionnement et de son mode d'interaction avec les autres objets ou les autres éléments — permet à l'utilisateur peu expérimenté d'écrire facilement des programmes.

La plupart des ordinateurs domestiques disposent d'une capacité mémoire et d'une puissance de traitement suffisantes pour répondre aux besoins de l'utilisateur. Toute augmentation de cette capacité et de cette puissance ne sert qu'à faciliter l'emploi de la machine. Ce qui est particulièrement remarquable à propos du PCS, c'est qu'il parvient à offrir une réelle facilité d'utilisation avec 48 K seulement de mémoire.

Alors que la programmation orientée vers les objets s'applique naturellement aux program-

mes de jeux et de graphisme, les programmeurs doivent faire preuve d'un peu plus d'ingéniosité pour l'introduire dans le domaine du logiciel de gestion.

Le système Lisa d'Apple, qui utilise une souris pour déplacer un indicateur sur l'écran afin de choisir diverses fonctions (représentées par un symbole graphique), est un autre exemple de ce type de programmation. Par exemple, le programme de traitement de texte est représenté par une feuille de papier à lettre, le programme graphique par une feuille de papier carrée.

Une des fonctions les plus fascinantes de l'ordinateur Lisa est le transfert de données d'un programme à un autre. Une de ses « icônes » (nom donné aux représentations picturales des fonctions) est un support à documents. Si l'utilisateur désire reproduire une portion d'une feuille de calcul électronique sur un graphique, il n'a qu'à définir une fenêtre sur la feuille de calcul, transférer cette fenêtre sur le support à documents (qui représente en fait une zone de stockage temporaire) et l'entraîner jusqu'au programme graphique.

Lorsque nous avons traité les jeux d'arcades (voir page 221), nous avons noté qu'ils formaient plusieurs types de programmes. Le PCS pourrait très bien créer une nouvelle catégorie. On peut supposer que les concepteurs de jeux électroniques proposeront bientôt des programmes encore plus variés.



Droit au but

En plus d'être captivant et éducatif, le programme Pinball Construction Set est un bon exemple de programmation orientée vers les objets.

En programmation normale, la structure des données est définie, et les routines du programme sont écrites afin qu'elles puissent manipuler ces données. En programmation orientée vers les objets, les calculs, les procédures et les données forment un tout indissociable. Dans le programme PCS déplacer le symbole d'un flipper sur la table ne fait pas qu'installer les données concernant le dispositif (la forme du flipper), mais transfère également les routines qui lui sont associées pour permettre son fonctionnement. La programmation orientée vers les objets s'adresse naturellement aux applications visuelles. Les tableaux forment un autre exemple : le champ renferme également la formule qui calcule ce résultat. (Cl. Chris Stevens.)



Missiles de croisière

Les missiles de croisière sont sujets à polémique, mais ils utilisent une technologie informatique — comme les mémoires à bulles — qui fera bientôt son apparition dans les ordinateurs domestiques.

C'est surtout grâce aux systèmes de guidage par ordinateur que l'homme a pu se poser sur la Lune. Ce fut le cas lorsque Neil Armstrong fit son premier petit pas sur le satellite de la Terre. Évidemment, ce type de liaison interplanétaire repose sur des équipements mécaniques très précis. Mais, sans la présence de matériel et de logiciel informatiques, les calculs de positionnement n'auraient jamais pu être effectués assez rapidement ou avec une précision suffisante pour permettre à un objet d'atteindre un autre objet à une distance aussi grande — même un objet de la dimension de la Lune.

Les normes militaires exigent qu'une ogive puisse avoir une précision de 20 à 30 m après un vol de plusieurs milliers de kilomètres. Voilà pourquoi une puissance de traitement énorme est au départ nécessaire pour effectuer les calculs aboutissant à une telle précision.

Les premières expériences militaires impliquant l'utilisation de missiles ont mis en évidence l'impossibilité de corriger la trajectoire des missiles après leur mise à feu. Un premier pas important fut franchi avec la mise au point de systèmes de guidage simples qui étaient en mesure d'évaluer la position du missile par rapport à un point de la surface du sol (la cible), en déduisant la distance et la direction du parcours. Mais même un système informatique moderne des plus sophistiqués sera alors exposé à des erreurs significatives.

Une autre méthode plus précise utilise des satellites géostationnaires comme points de référence. Le principal inconvénient de ce système est que la trajectoire — et probablement l'objectif — du missile peut être déduite rapidement après le lancement par l'ennemi, grâce au potentiel des systèmes radar modernes. Pour éliminer cette vulnérabilité, il était nécessaire de mettre au point un missile volant à très basse altitude muni d'un système radar capable de définir de façon continue la trajectoire de vol vers l'objectif. C'est ainsi qu'est né le missile de croisière.

Le missile de croisière met constamment à jour sa position en analysant le relief du terrain survolé. Pour cela, il compare les lectures d'altitude au-dessus du sol effectuées par un altimètre radar extrêmement précis avec une carte topographique stockée dans une mémoire à bulles intégrée.

Ce système mis au point par McDonnell Douglas a été baptisé TERCOM (*TERrain COntour Matching*), ou DPW-23. Chaque missile stocke dans sa mémoire à bulles quelque vingt-cinq



« profils de trajectoire », qu'il compare avec le terrain survolé. Ce système présente cependant des inconvénients.

Le missile ne commence pas à utiliser ce système de guidage dès son lancement. Il demeure inactif tant que le missile vole à haute altitude dans un espace aérien allié. Dès qu'il y a risque d'attaque terrestre ou aérienne, il plonge à 15 m au-dessus du sol pour survoler le territoire ennemi. Même si le décalage par rapport à la trajectoire prévue atteint 1 km à ce moment, il est prévu qu'il sera suffisamment près de l'un des tracés stockés dans sa mémoire pour être capable de localiser précisément sa position.

Lorsque le missile est près de son objectif, il met en fonction un dispositif d'approche finale qui renferme — de nouveau en mémoire à bulles — une image numérique détaillée de la zone où se situe l'objectif, telle qu'elle apparaîtrait vue d'un missile en approche. Les essais ont démontré que le point d'impact de ce missile se situerait à moins de 18 m de la cible après un vol de 2 800 km.



Missile à tête chercheuse

Le missile de croisière Tomahawk de la General Dynamics a une longueur de 6,40 m et pèse moins de 1 200 kg. Mis à feu à partir d'un lanceur mobile, sa trajectoire est d'abord celle d'une fusée conventionnelle, mais il déploie bientôt de petits ailerons et descend à faible altitude où il est propulsé par un moteur à réaction à turboventilation remarquablement petit. (Cl. Aerospace Publishing Ltd.)

Bulles, bulles

Dans les mémoires à bulles, des « bulles » magnétiques sont créées pour former un 1, et leur absence spécifie un 0, sur une minuscule puce de grenat. Les avantages sont la densité de stockage — 1 million de bits ou 128 K par puce — et aucune perte du contenu lors de mise hors tension. (Cl. New Scientist.)



Listes enchaînées

L'indexation est une façon de structurer de grandes quantités de données, comme des listes de noms et d'adresses. On peut également utiliser des listes enchaînées.

Données en mémoire

L'organisation d'une base de données correspond à l'implantation des données en mémoire et à celle des relations qui peuvent exister entre ces données. (Cl. Gayard/Rea.)

La mémoire de l'ordinateur ne renferme que des données qui sont stockées octet après octet sous la forme de milliers de configurations binaires. La signification de ces octets est dictée par la structure des données imposée par l'unité centrale de traitement. Les diverses structures de données déterminent la nature de chaque octet ;

un octet peut désigner une partie d'une instruction, un chiffre faisant partie d'un nombre, ou encore un code de caractère.

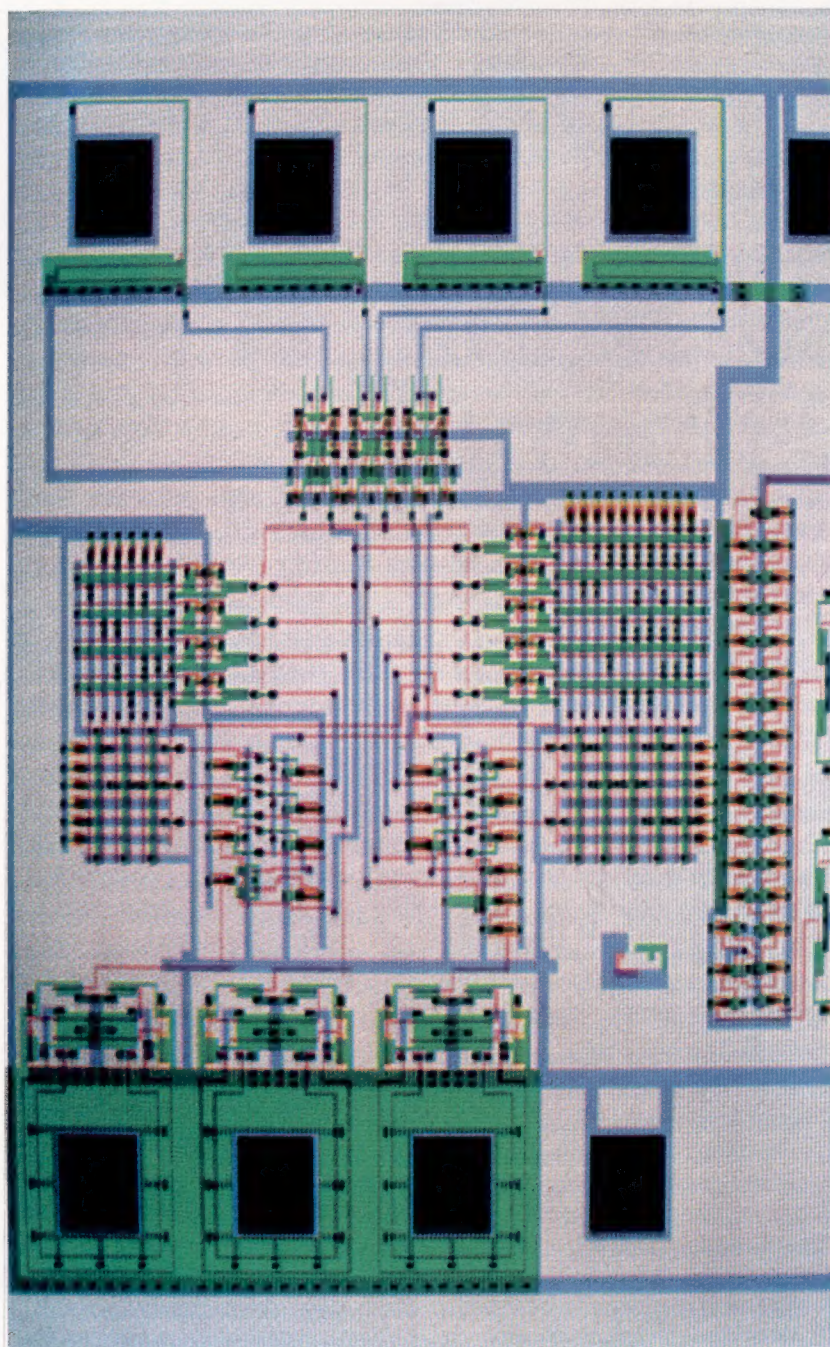
Du point de vue de l'utilisateur, certains types de structures de données sont en fait définis de façon permanente dans l'ordinateur. Les langages de programmation demandent généralement que les données soient structurées dans un nombre limité de possibilités. Le BASIC impose le stockage de données dans des variables numériques et dans des variables de chaîne, et offre des structures de variables et de tableaux pour les manipuler. D'autres langages offrent également ces structures et d'autres supplémentaires. La puissance d'un langage est fonction du nombre et de la variété des types de données.

Les tableaux indexés constituent une structure de donnée pratique et facile à mettre en œuvre en BASIC. Elle est cependant limitée, particulièrement lorsque les données auxquelles elle se réfère sont susceptibles d'être modifiées fréquemment et/ou de façon imprévisible.

Supposons que le service éditant les annuaires téléphoniques détienne un fichier de ses nouveaux abonnés en vue d'une éventuelle inclusion dans la prochaine édition. Les noms et adresses doivent être stockés par ordre alphabétique afin de faciliter leur consultation. Mais le fichier ne cesse de croître et les additions surviennent de façon imprévisible. Le lundi le fichier `NouvAb$()` peut se présenter ainsi en mémoire :

NouvAb\$()	Index ()
(1) Jullien	(2)
(2) Avenel	(3)
(3) Camy	(6)
(4) Ronsard	(1)
(5) Severin	(4)
(6) Delavaud	(5)

Le tableau `Index()` illustre dans quel ordre lire `NouvAb$()` afin que les entrées soient classées alphabétiquement. Par conséquent, le premier élément dans un classement alphabétique est `NouvAb$(2)`, Avenel. Le deuxième élément est `NouvAb$(3)`, Camy. Seuls les noms sont donnés dans cet exemple, mais une entrée d'annuaire comprend en fait le nom, les initiales et l'adresse, soit environ une soixantaine de caractères. Le déplacement de soixante caractères en mémoire est lent (puisque le tri nécessite de nombreux déplacements de données) et occupe beaucoup d'espace mémoire. Il est donc plus efficace de





ne pas trier `NouvAb$()`, et de créer `Index()` à la place. Maintenant, un nouveau nom, `Belair`, doit être ajouté au fichier. Les tableaux se présentent alors ainsi :

NouvAb\$()	Index()
(1) Jullien	(2)
(2) Avenel	(7)
(3) Camy	(3)
(4) Ronsard	(6)
(5) Severin	(1)
(6) Delavaud	(4)
(7) Belair	(5)

Notez que les contenus de `Index()` au-dessus de la nouvelle insertion n'ont pas changé et que les éléments au-dessous de l'insertion sont restés dans le même ordre, mais ont tous été décalés d'une position vers le bas. L'insertion d'un nouvel élément implique donc : trouver la position du nouvel élément, décaler d'une position vers le bas chaque élément compris entre cette position et la fin du fichier, et écrire la nouvelle entrée. Il est préférable de faire cela plutôt que de trier les véritables données, `NouvAb$`.

Supposons maintenant que nous structurions les données d'une manière différente. Ne trions pas `NouvAb$()` parce que sa manipulation est trop longue, et établissons un tableau parallèle nommé `Rech()` qui ne renferme que des nombres désignant les positions dans `NouvAb$()`.

TêteListe(2)

NouvAb\$()	Rech()	Index()
(1) Jullien	(4)	(2)
(2) Avenel	(3)	(3)
(3) Camy	(6)	(6)
(4) Ronsard	(5)	(1)
(5) Severin	(0)	(4)
(6) Delavaud	(1)	(5)

On a besoin d'une variable simple nommée `TêteListe` : elle désigne `NouvAb$(2)` qui est alphabétiquement le premier élément de `NouvAb$()`. De plus, le nombre (0) a été utilisé dans `Rech(5)` : cela indique que `NouvAb$(5)` est le dernier élément du tableau classé par ordre alphabétique.

La différence suivante est le contenu de `Index()` et de `Rech()`. `Index` doit être lu ainsi : « Le premier élément est dans `NouvAb$(2)`, le deuxième élément est dans `NouvAb$(3)`, le troisième élément est dans `NouvAb$(6)`, etc. » Alors que `TêteListe()` est lu : « Le premier élément est dans `NouvAb$(2)`, puis `Rech(2)` indique que le prochain élément est dans `NouvAb$(3)`; `Rech(3)` indique que le prochain élément est dans `NouvAb$(6)`, et ainsi de suite. » `Rech(5)` indique que `NouvAb$(5)` est le dernier élément.

`Index()` donne une position absolue des éléments du fichier, alors que `Rech()` ne donne que des positions relatives — tout élément dans `Rech()` n'indique que l'endroit où trouver le prochain élément mais ne dit pas sa position absolue. Le nombre dans `Index(4)` désigne le quatrième élé-

ment dans un fichier disposé alphabétiquement, tandis que le nombre dans `Rech(4)` ne désigne que l'élément qui suit `NouvAb$(4)` dans le fichier ordonné. `Rech()` met en place une structure de données dite « liste enchaînée ». Lire une liste enchaînée ressemble un peu à une course au trésor : au début on vous indique votre première destination, à cet endroit vous trouvez un indice qui vous dirige vers la destination suivante, et ainsi de suite.

Le grand avantage de la structure de liste est sa souplesse. Examinons la nouvelle liste.

TêteListe (2)

NouvAb\$()	Rech()
(1) Jullien	(4)
(2) Avenel	(7)
(3) Camy	(6)
(4) Ronsard	(5)
(5) Severin	(0)
(6) Delavaud	(1)
(7) Belair	(3)

Le tableau `Rech()` n'a été modifié qu'à deux endroits :

a) `Rech(2)`, qui précédemment désignait `NouvAb$(3)` comme prochain élément alphabétique après `NouvAb$(2)`, désigne maintenant `NouvAb$(7)` puisqu'il est maintenant l'élément qui suit `NouvAb$(2)` en ordre alphabétique.

b) `Rech(7)`, qui n'était pas utilisé, désigne maintenant `NouvAb$(3)` comme l'élément qui suit `NouvAb$(7)` en ordre alphabétique.

Voici le processus général d'insertion dans une liste enchaînée : trouver l'élément de la liste qui doit immédiatement précéder le nouvel élément, faire en sorte qu'il désigne le nouvel élément; puis s'assurer que le nouvel élément désigne l'élément qu'il a déplacé. Ces deux opérations simples suffisent pour insérer un élément dans une liste enchaînée. Insérer un élément dans une liste c'est comme introduire un nouveau maillon dans une chaîne — décider où mettre le maillon, briser la chaîne, joindre le maillon précédent au nouveau maillon, et le nouveau maillon au maillon suivant. Les nombres contenus dans `Rech()` — les maillons — sont souvent nommés des « pointeurs ».

Une des caractéristiques importantes des listes enchaînées est leur caractère séquentiel marqué; il est impossible de trouver un élément dans une liste autrement qu'en commençant au début et en examinant chaque élément jusqu'à ce que l'élément cible soit trouvé. La liste est ici mise en place à l'aide de tableaux, qui sont au départ des structures d'accès direct. Mais la liste les a transformés en fichiers séquentiels.

Les listes sont des structures pratiques pour manipuler des données dynamiques (données qui changent régulièrement), et peuvent être des outils puissants pour le traitement du langage naturel (comme en reconnaissance de la parole) ou du langage artificiel (compilation de programme) où les données forment naturellement une liste d'éléments.



Présentation du son...

« Son et lumière » est une nouvelle série d'articles où vous apprendrez comment utiliser au maximum le potentiel graphique et sonore de votre ordinateur.

Les ordinateurs domestiques ont bénéficié de nombreuses améliorations au cours de ces dernières années; ils deviennent de plus en plus polyvalents. La popularité d'un nouvel ordinateur dépend toujours de la richesse technique des jeux qu'il peut offrir. Les constructeurs se sont donc efforcés de développer considérablement le potentiel graphique couleur de leurs machines. Bien que leur importance ne soit pas aussi évidente, les fonctions sonores et musicales ont suivi le mouvement. Si vous demandiez à un auteur de programmes de jeux quelle importance il accorde aux routines sonores dans ses jeux, il les placerait probablement en troisième position, juste derrière la conception et le

graphisme du programme. Un usage judicieux des effets sonores et de musique améliore considérablement l'impact et la qualité de tous les types de jeux d'arcades.

En plus des applications ludiques, vous pouvez approfondir vos connaissances musicales en utilisant le potentiel sonore de votre ordinateur domestique. De nombreuses versions de BASIC offrent des fonctions musicales qui vous permettent d'écrire de courts programmes — comportant même des accords — pour exécuter des airs assez complexes. Certains ordinateurs permettent également de modifier la nature du son produit pour le rendre plus harmonieux ou pour imiter les sons de divers instruments de musique. Dans tous les cas, le clavier peut être arrangé, au moyen d'un programme approprié, en un clavier de piano qui vous permet de jouer de la musique « en temps réel ».

Avec un minimum de notions de programmation, il est possible d'écrire de courts et simples programmes qui produisent des morceaux de musique assez évolués. Si vous désirez exploiter au maximum le potentiel sonore de votre ordinateur, plusieurs sociétés de logiciel produisent des programmes musicaux très complets qui vous permettent d'écrire et d'exécuter directement des morceaux musicaux. Quelle que soit l'approche que vous retenez, il est judicieux de comprendre comment votre ordinateur génère, définit et commande sa sortie sonore.

...et de l'image

Faible et haute résolutions

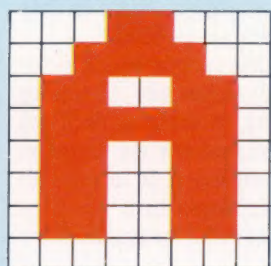
Les graphismes sur micro-ordinateurs peuvent être divisés en deux catégories : faible et haute résolutions. Pour mieux décrire la différence existant entre ces deux types de graphisme, examinons comment est formé un caractère.

Si vous examinez de près un caractère standard affiché à l'écran, vous verrez qu'il est composé d'un ensemble de petits carrés. Ces carrés sont les éléments de l'image, s'appellent des points d'affichage ou « pixels », et chaque caractère est formé par une configuration particulière de ces points. Sur la plupart des ordinateurs domestiques, les caractères sont formés à l'intérieur d'une grille de soixante-quatre points, disposés sur huit lignes de huit colonnes.

Chaque point illuminé sur la grille est représenté dans la mémoire de l'ordinateur par un 1 et chaque point sombre par un 0. Huit bits formant un octet, chaque ligne de la grille d'un caractère peut être stockée dans un seul emplacement de la mémoire de l'ordinateur. Chaque caractère occupe donc huit emplacements de mémoire.

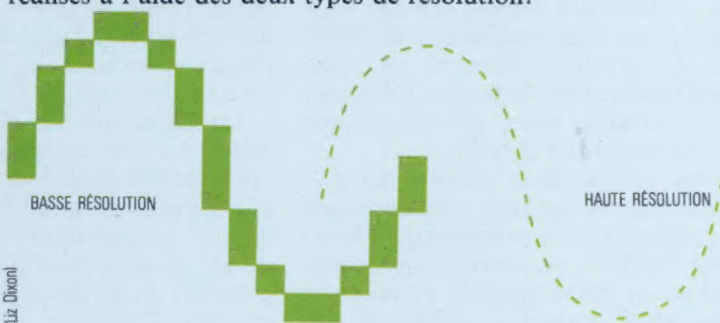
Les affichages graphiques sont quelquefois formés à l'aide de blocs dont la dimension équivaut à la totalité, à la moitié ou au quart d'une grille de caractère. Les graphismes conçus à l'aide de ces grands blocs sont appelés graphismes à faible résolution. De nombreux ordinateurs domestiques permettent de concevoir des graphismes construits à partir de points individuels. Présentons deux tracés sinusoïdaux réalisés à l'aide des deux types de résolution.

CONFIGURATION DE POINTS



CONFIGURATION DE BITS

0	0	0	1	1	0	0	0
0	0	1	1	1	1	0	0
0	1	1	0	0	1	1	0
0	1	1	1	1	1	1	0
0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	0
0	0	0	0	0	0	0	0



(Luz Dixon)



Oscillateurs

Les oscillateurs sont des circuits électroniques qui génèrent des signaux répétitifs. Lorsque ces signaux sont amplifiés et transmis à un haut-parleur, ils produisent des sons à un ton spécifique. Les ordinateurs domestiques possèdent de un à quatre oscillateurs.

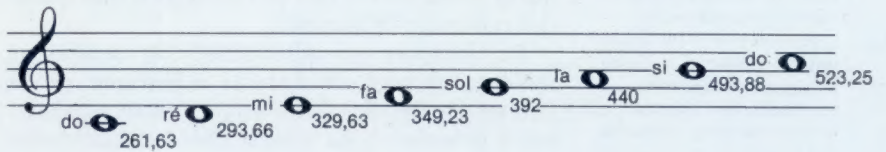
Trois caractéristiques définissent le son créé : la fréquence, l'enveloppe (qui comprend le volume) et la forme d'onde. La fréquence sera décrite dans cet article, tandis que les générateurs d'enveloppe et la forme d'onde seront traités ultérieurement.

Fréquence

C'est la caractéristique la plus importante du son que nous devons commander, puisqu'elle détermine la note produite. La fréquence désigne le nombre de répétitions d'un signal survenant à chaque seconde et est exprimée en hertz (Hz), ou cycles par seconde. L'oreille peut entendre les sons dont les fréquences sont supérieures à 20 Hz et inférieures à environ 20 000 Hz. Bien que nous ne puissions entendre les fréquences inférieures à 20 Hz, elles peuvent servir à modifier les caractéristiques d'un son audible. Cette technique s'appelle modulation, et ne peut actuellement être appliquée que sur le

Commodore 64 parmi les ordinateurs domestiques du marché.

Il n'est cependant pas nécessaire d'approfondir la notion de fréquence. Ce que vous devez savoir, c'est comment produire les diverses notes musicales. La facilité de sélection des notes varie énormément d'une machine à l'autre. Certaines machines ont des commandes BASIC qui définissent automatiquement les fréquences, vous n'avez qu'à spécifier le nombre correspondant ou la note elle-même : *do*, *do #*, *ré*, etc. D'autres ne fournissent qu'un tableau dans le manuel de l'utilisateur, où vous devez trouver la fréquence correspondant à la note désirée; vous devez ensuite écrire (à l'aide de l'instruction POKE) la valeur de la fréquence dans une adresse de la mémoire. L'illustration ci-contre donne des conversions précises pour la gamme située au-dessous du *la* du diapason. Ces valeurs sont très pratiques pour ceux qui désirent programmer de la musique en langage machine, où le BASIC ne peut vous aider à calculer les fréquences.



Des notes aux fréquences
Vous pouvez calculer la fréquence de chaque note de la gamme en multipliant par 1,0594631 la fréquence de la note située un demi-ton plus bas. Cela peut sembler très peu, mais si la multiplication est effectuée à douze reprises, la fréquence initiale est doublée. Il y a douze demi-tons dans une octave. Doubler la fréquence fait donc monter le son d'une octave. Cette illustration donne les valeurs précises des fréquences correspondant aux notes de la gamme située sous le *la* du diapason.

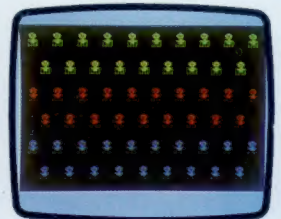
Caractères définis par l'utilisateur

Afin de créer des affichages inhabituels et intéressants, il est souvent utile de pouvoir utiliser des caractères autres que ceux qui sont offerts par le jeu de caractères alphanumériques standard. Le Vic-20 et le Commodore 64 ont un jeu de caractères spécial qui peut être utilisé directement au clavier. Mais même celui-ci ne peut répondre à tous les besoins. Il est possible de créer de nouveaux caractères sur la plupart des ordinateurs domestiques. Cela est généralement réalisé en redéfinissant les configurations binaires de huit adresses de mémoire où le caractère est stocké. Au cours de ce processus, l'ancien caractère est souvent perdu, ou « écrasé », et ce caractère « défini par l'utilisateur » adopte certaines des propriétés du caractère qu'il a remplacé en mémoire. Ainsi, le nouveau caractère peut être utilisé dans des instructions PRINT, en pressant simplement la touche correspondant au caractère qu'il a remplacé. Voici un exemple d'un caractère défini par l'utilisateur :

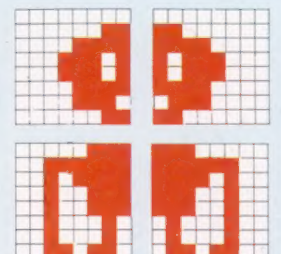
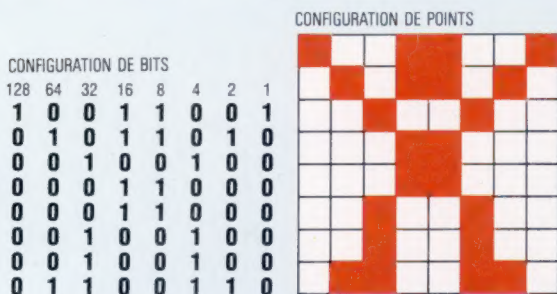
La difficulté de créer ces caractères personnalisés varie considérablement selon l'ordinateur utilisé. Par exemple, avec la commande USA du Sinclair Spectrum, il suffit d'entrer les configurations binaires appropriées; tandis que, sur le Commodore 64, l'utilisateur doit d'abord déplacer le jeu de caractères complet de la ROM à la RAM avant de pouvoir écrire dans la mémoire, à l'aide de l'instruction POKE, les huit nombres décimaux correspondant aux configurations de bits qui définissent le symbole désiré.

Pour créer de plus grands symboles, il est possible de grouper deux ou plusieurs caractères. Les personnages extra-terrestres représentés à droite ont été construits avec quatre caractères personnalisés. Ce programme est exécuté sur le Commodore 64, et affiche les groupes de caractères en trois couleurs différentes. Les caractères ont été créés à l'aide d'une courte routine qui chargea en RAM le jeu de caractères complet contenu en ROM et qui remplaça les caractères graphiques ■, □, ▣, et " " par des valeurs lues dans des instructions DATA et écrites à l'aide de commandes POKE aux adresses appropriées. Nous décrirons ce processus de façon détaillée dans un article ultérieur.

Même lorsque des plans objets (voir page 152) sont disponibles, le nombre d'objets pouvant être affichés simultanément est souvent limité. Les graphismes définis par l'utilisateur peuvent ainsi être très utiles pour afficher simultanément de nombreuses formes similaires.



Extraterrestres
Ces étranges personnages furent créés à l'aide de quatre caractères définis par le programmeur. (Cl. Ian McKinnell.)





Meilleures prévisions

L'utilisation d'ordinateurs très rapides, pour le traitement des images obtenues par satellite et pour l'analyse des données, permet d'obtenir des prévisions météorologiques beaucoup plus précises.

Images de l'espace

Le satellite météorologique Météosat 2, lancé en juin 1981, est placé sur une orbite géostationnaire (c'est-à-dire qu'il ne bouge pas par rapport à la Terre), à quelque 35 880 km au-dessus de l'équateur, sur le méridien zéro. Il récolte des informations provenant de nombreuses stations terrestres.

Sans nous en rendre compte, nous profitons souvent dans notre vie quotidienne des progrès réalisés dans le traitement de l'information. Une des applications informatiques les plus évoluées, nécessitant une énorme capacité de traitement, permet de fournir des informations concernant les conditions météorologiques et leur évolution probable. La complexité de la prévision dans ce domaine est telle qu'il peut paraître surprenant que nos météorologistes réussissent aussi souvent à faire de bonnes prévisions. L'ordinateur représente pour eux un atout considérable lors de l'analyse d'une quantité colossale de données.

Les facteurs qui déterminent les conditions météorologiques du littoral atlantique de la France (et des îles Britanniques) sont extrêmement complexes. Ils sont d'abord conditionnés par la proximité de cette région par rapport au pôle Nord et à l'océan Atlantique. Étant située

du côté est de l'Atlantique, cette zone est exposée aux conditions climatiques créées sur une largeur de 4 000 km, en raison de l'« effet Coriolis ». Ce phénomène est dû à la rotation terrestre ouest-est. Pour mieux le comprendre, rappelons qu'un objet se trouvant à l'équateur au niveau du sol se déplace à plus de 1 600 km/h; ce mouvement de rotation puissant, combiné aux vents normaux dans le sens pôle-équateur, crée une prédominance des vents d'ouest dans l'hémisphère Nord. Cette arrivée constante d'air humide — ascendant ou descendant selon les températures locales — cause les conditions climatiques particulières à l'ouest de la France et à la Grande-Bretagne.

Les météorologistes se fondent principalement sur des observations provenant de diverses stations météorologiques placées à des endroits stratégiques dans l'Atlantique — navires météorologiques, bouées, ballons et avions. Grâce à ces informations, ils peuvent prédire ce qui arrivera quand les conditions climatiques observées





atteindront le continent européen, en se fondant sur le comportement connu de phénomènes similaires appartenant au passé.

Avant le lancement du satellite météorologique Météosat 1, le 19 mars 1979, la seule méthode de prévision dont disposaient les spécialistes consistait à rapporter sur une carte les observations des stations météorologiques et à tracer des courbes isobares. Les courbes isobares sont des lignes imaginaires qui relient des points de même pression atmosphérique, un peu comme les courbes de niveau relient sur une carte les points de même élévation. Grâce à ces courbes, il est possible de déduire la vitesse et la direction des fronts chauds et froids — et des cyclones et anticyclones associés — et ainsi de tenter une prévision sur ce que seront les conditions météorologiques.

La Météorologie nationale se sert encore de ce procédé pour obtenir des représentations précises des conditions en cours, mais utilise également les images fournies par Météosat. Celles-là



Le traitement de ces photos obtenues par satellite n'est que l'une des nombreuses fonctions du système informatique de la Météorologie nationale. Avec la collaboration d'autres organismes similaires dans d'autres parties du monde, elle met continuellement à jour un modèle météorologique global et extrait de ce modèle une quantité importante de données statistiques, qui constituent une base de données, à

Stations terrestres

Les antennes de réception satellite (nommées antennes paraboliques) sont de dimensions et de complexité extrêmement variables. Cette station transmet et reçoit des signaux vers et en provenance de satellites géostationnaires.

0 2 5 4 5 7 6 9
3 2 1 9 8 3 4 0

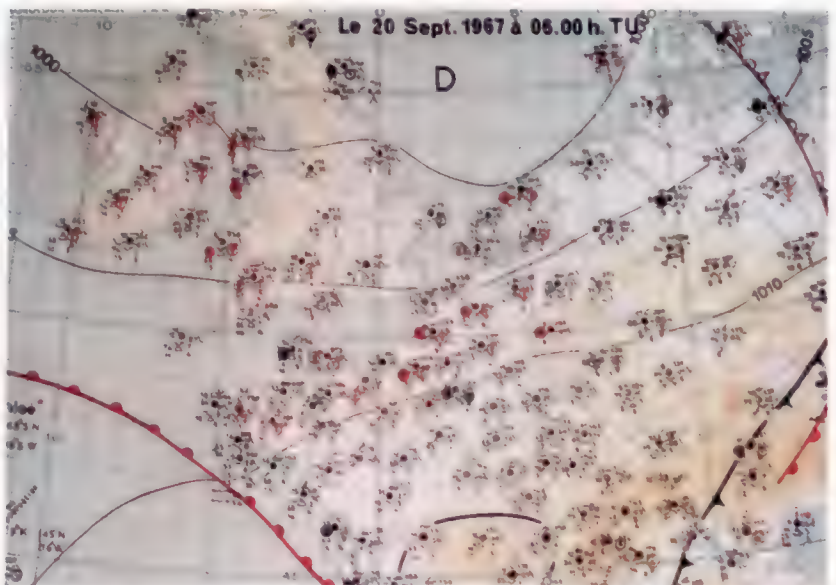
Broyeurs de nombres

L'une des tâches les plus importantes des gros ordinateurs utilisés en recherche scientifique est de traiter des informations essentiellement numériques sous la forme d'équations longues et complexes. Les recherches purement scientifiques comme la physique nucléaire et les sciences appliquées comme la météorologie ont des exigences similaires. Bien qu'il soit possible d'effectuer des calculs de cette complexité sur un ordinateur domestique, cela serait beaucoup trop long, non seulement en raison du nombre de variables mais aussi de la grandeur des nombres.

sont envoyées sous la forme de signaux analogiques qui sont numérisés, traités et affichés par l'ordinateur sous la forme de cartes colorées artificiellement. Les images créent une représentation vivante des conditions météorologiques. Elles sont régénérées toutes les quatre minutes, ce qui permet donc de suivre l'évolution météorologique en temps réel.

Météosat 2, qui remplaça Météosat 1 en juin 1981, évolue sur une orbite géostationnaire à quelque 35 880 km au-dessus de l'équateur. Il récolte des données provenant de nombreuses stations terrestres disséminées sur la surface du globe, et transmet ces informations à tous les organismes désirant adhérer au système international de météorologie.

Il serait théoriquement possible d'analyser et d'interpréter cette information (pas en temps réel) sur un ordinateur domestique en écrivant les données reçues sur une disquette pendant qu'elles sont envoyées par le satellite. Cependant, le signal étant analogique, la conversion pourrait être difficile. Vous devriez également installer votre propre antenne parabolique alignée de façon précise vers le satellite.



l'aide de laquelle il est possible de tracer les tendances climatiques globales et locales.

La collecte de ces données est importante pour l'analyse historique. Ces informations sont vitales pour les agriculteurs, pour plusieurs industries, pour l'économie et l'écologie de continents entiers, puisque c'est uniquement par ce moyen que les changements de climat peuvent être reconnus. C'est ainsi que l'on a pu repérer la destruction progressive de la forêt tropicale amazonienne et la progression de la calotte polaire. Ces phénomènes pourraient indiquer l'approche d'une période glaciaire.

Courbes isobares

Les vents circulent dans le sens inverse des aiguilles d'une montre autour d'une zone de basse pression et dans le sens des aiguilles d'une montre autour d'une zone de haute pression (c'est l'inverse dans l'hémisphère Sud), et la vitesse du vent est directement proportionnelle à la distance entre les courbes isobares. (Cl. Météorologie nationale.)



Sord M5

Bien que cette machine n'offre que 4 K de mémoire, ses puissantes fonctions graphiques permettent à l'utilisateur d'écrire des programmes valables.

La plupart des premiers ordinateurs domestiques furent conçus en Californie. Plus récemment, des machines britanniques ont commencé à prendre une vaste part du marché mondial. Il est fort probable cependant que les Japonais ne tarderont pas à dominer ce secteur d'activité, comme ils l'ont fait dans tous les autres marchés d'appareils électroniques destinés au grand public. Le Sord M5 n'est certainement pas le premier micro-ordinateur japonais, mais il est le premier à s'attaquer au marché domestique et non au marché professionnel.

Il possède un boîtier solide et compact, de dimensions comparables à celles du Sinclair Spectrum, mais il est beaucoup plus lourd et semble beaucoup plus robuste. Il est doté de caractéristiques similaires à celles du Spectrum à plusieurs points de vue, avec une UC Z80A, des entrées à une touche pour fonctions BASIC, et un stockage de programmes et de données sur cassette. Son fonctionnement interne est cependant beaucoup plus sophistiqué, comme le démontre la présence du port d'imprimante Centronics intégré. Mais deux différences essentielles apparaissent : la taille de la mémoire RAM (extension possible jusqu'à 36 K) est beaucoup plus petite dans la version de base; la présence de puces graphiques et sonores dédiées. Deux différences qui donnent un attrait particulier au Sord M5.



La cartouche ROM

L'une des caractéristiques les plus intéressantes du Sord M5 est que le langage peut être changé puisqu'il est contenu dans une cartouche ROM. Trois versions de BASIC sont offertes sur le M5 : BASIC-1 (simple pour débutant); BASIC-G (avec fonctions graphiques très évoluées); BASIC-F (destiné aux applications scientifiques et mathématiques). Il existe aussi un programme universel nommé FALC.

Borne imprimante

Une interface parallèle compatible Centronics peut être placée dans cette prise, ce qui permet de connecter au Sord M5 l'une des nombreuses imprimantes disponibles sur le marché.

Borne RF

La sortie compatible téléviseur est placée ici.

Modulateur

La sortie provenant du VDP est convertie en un signal télévision standard.

Borne vidéo

Le signal vidéo composite non modulé fourni à cet endroit peut être transmis à un moniteur.

Borne audio

La sortie audio peut être transmise à un amplificateur à partir de ce connecteur.

VDP

Le processeur d'affichage vidéo Texas TMS 9929 est responsable de l'affichage et peut gérer jusqu'à 32 plans objets séparés.

Bornes commandes manettes

Les deux commandes manettes sont connectées à cet endroit.

RAM vidéo

Toutes les données nécessaires pour gérer l'affichage, y compris les images, sont contenues dans ce bloc de RAM de 16 K.



Borne d'unité à cassette

L'interface cassette est une prise de type DIN, et possède les connexions nécessaires pour commander le moteur de l'unité à cassette.



Les boîtiers de commandes

Les « joypads » sont les manches à balai du Sord. Ils fonctionnent en envoyant un signal pour chacune des quatre directions diagonales. Puisque ces signaux interrompent l'UC, le temps de réponse est très rapide.

Borne d'alimentation

Présence d'un petit transformateur.

Puce personnalisée

Le M5 utilise une puce logique très sophistiquée, ce qui lui permet d'offrir à un prix abordable des fonctions évoluées.

ROM

Les seuls programmes intégrés dans cette machine sont un ensemble de programmes de commande de bas niveau qui sont appelés par le programme utilisateur. Ceux-ci se chargent des détails de gestion de l'écran, du clavier et de la cassette.

UC (CPU)

Le microprocesseur du Sord M5 est le célèbre Z80A. Celui-ci est synchronisé à 3,58 MHz.

RAM

La mémoire utilisateur est contenue dans ces deux puces, et est séparée des autres zones de RAM.

CTC

L'efficacité du fonctionnement du M5 est surtout due à ce contrôleur d'horloge évolué qui synchronise et sollicite les diverses interventions de l'ordinateur.

Les fonctions graphiques sont gérées par une puce TI 9918, 9928 ou 9929 (selon le pays où l'ordinateur est vendu); l'affichage possède une résolution de 192×256 points en seize couleurs au maximum. Il existe quatre principaux modes graphiques, dont trois offrent jusqu'à trente-deux plans objets à déplacement indépendant qui peuvent être de dimensions standard ou agrandis. La machine peut afficher des lettres minuscules et majuscules, les ponctuations et les chiffres. Elle peut aussi afficher des symboles permettant de tracer des encadrés et des lignes, ainsi qu'un vaste choix de lettres minuscules accentuées qui permettent d'écrire dans plusieurs langues. En outre, tout caractère pouvant être redéfini, les possibilités sont très larges.

D'autres ordinateurs utilisent les mêmes puces graphiques — particulièrement le

**SORD M5****PRIX**

1 800 F

DIMENSIONS

185 x 70 x 55 mm

POIDS

1 kg

UC

Z80A

VITESSE DE L'HORLOGE

3,58 MHz

MÉMOIRE

8 K de ROM, 20 K de RAM, dont 16 K sont réservés à l'affichage graphique. Avec des cartouches supplémentaires, la ROM peut être portée à 16 K et la RAM à 32 K.

AFFICHAGE VIDÉO

Jusqu'à 16 couleurs qui peuvent être utilisées dans des « plans » différents. Il y a des plans objets et quatre modes d'affichage différents : deux modes graphiques, un mode texte et un mode « multicolore ».

INTERFACES

Cassette, imprimante (Centronics), « joypads », cartouche ROM, audio.

LANGAGE FOURNI

La cartouche langage fournie est le BASIC-1.

AUTRES LANGAGES DISPONIBLES

BASIC-G (graphiques), BASIC-F (BASIC à virgule flottante), FALC.

ACCESSOIRES FOURNIS

Adaptateur d'alimentation, fils de connexion de cassette, fil de connexion de téléviseur, deux joypads avec fils de connexion, cartouche BASIC-1 et une cassette contenant deux jeux.

CLAVIER

55 touches : huit fonctions de commande définissent tous les caractères alphanumériques, 28 instructions BASIC et 64 configurations graphiques.

DOCUMENTATION

Le guide de l'utilisateur compte 18 pages et décrit comment connecter l'ordinateur, comment charger et exécuter les deux jeux, et comporte une page de conseils de dépannage.

TI-99/4A (voir page 189) — et c'est la présence de telles puces dédiées qui permet au Sord M5 d'être si efficace, malgré la faible taille de sa RAM. Puisque la mémoire écran est entièrement séparée de la mémoire programme, seul le programme est logé en RAM.

Un vaste débat dans l'industrie de l'ordinateur domestique concerne l'établissement d'une norme de fabrication, nommée « norme MSX », développée par un groupe de fabricants japonais dont Sord fait partie. Il s'agit de proposer aux fabricants le respect de cette norme dans la conception des ordinateurs domestiques (en ce qui concerne le matériel et le BASIC intégré), ce qui permettrait d'écrire des logiciels qui fonctionneraient sur toutes les machines, sans modification. Au niveau des puces graphiques, le Sord M5 obéit à cette norme.

Mais la norme MSX spécifie également que la puce sonore doit être la puce AY-3-8910 de General Instruments. Pour produire des sons, le Sord M5 (comme le micro BBC) utilise une puce TI 76489, qui permet une meilleure commande des sons que la puce de General Electric, bien qu'elle soit de conception similaire avec trois canaux sonores et un canal de bruit. Cela signifie que le Sord M5 n'est pas une véritable machine MSX.

Le clavier du Sord M5

Ce clavier caoutchouc est légèrement plus grand que celui du Sinclair Spectrum, et son toucher, plus sensible, rend la frappe plus agréable. 55 touches au total peuvent être utilisées de manières différentes afin d'obtenir des caractères alphanumériques, des symboles graphiques, ou des mots clés BASIC, à l'aide de la touche FUNC. Toutes les touches sont répétées automatiquement lorsqu'elles sont maintenues enfoncées, ce qui est très pratique pour l'édition à l'écran. (Cl. Chris Stevens.)



Contre les impairs

La « parité » assure que le nombre de bits mis à 1 dans un octet est toujours pair. Cela permet de détecter les erreurs de transmission.

Un des principaux avantages des ordinateurs numériques par rapport aux ordinateurs analogiques est que les erreurs et les imprécisions, inhérentes à tout circuit électrique, ne peuvent s'accumuler lorsqu'un signal traverse une multitude de circuits (voir page 239). Cependant, quand les données sont transmises à distance, soit au moyen d'une interface série et de deux paires de fils, soit *via* le réseau téléphonique, le « bruit » de fond de la ligne de transmission peut quelquefois transformer un 1 en un 0, ou vice versa. Normalement, l'ordinateur récepteur ne pourrait pas savoir que cela est arrivé, et accepterait les données erronées comme des données correctes.

Regardez ce qui se passe si un bit du code ASCII de la lettre Q est inversé :

[] 1 0 1 0 0 0 1 (code ASCII transmis pour la lettre Q)

[] 1 0 0 0 0 0 1 (code ASCII reçu pour la lettre A)

Une telle erreur de transmission de données serait pour le moins gênante ou pourrait se révéler catastrophique. Cependant, rappelons que les codes ASCII ne sont affectés qu'à des valeurs allant jusqu'à 127, ce qui ne nécessite que 7 bits (numérotés 0 à 6). Le bit le plus significatif (bit 7) est donc souvent utilisé comme bit de « parité », et sert à détecter les erreurs.

Il existe deux conventions d'utilisation des bits de parité : « parité » et « imparité ». Examinons la première convention. Dans une transmission en « parité », le bit de parité est mis à 1 ou à 0, de façon que le nombre total de bits mis à 1 soit toujours un nombre pair. Voici comment les lettres A et Q se présentent en parité :

[0] 1 0 0 0 0 0 1
(le code ASCII pour la lettre A en parité)

[1] 1 0 1 0 0 0 1
(le code ASCII pour la lettre Q en parité)

Deux bits sont mis à 1 dans le code ASCII de la lettre A; le bit de parité est donc mis à 0 afin que le nombre total de bits mis à 1 soit pair. Dans le code ASCII pour Q, il y a 3 bits mis à 1, le bit de parité est donc mis à 1. Cela donne un nombre total de 4 bits mis à 1, qui est un nombre pair.

Voyons maintenant ce qui serait survenu si le bit 4 de notre lettre ASCII Q avait été inversé lors de la transmission.

[1] 1 0 0 0 0 0 1 (code ASCII de Q altéré)

Lorsque la parité de cet octet est vérifiée (que ce soit par logiciel ou par matériel), une lettre Q

correcte renferme un nombre pair de 1 (incluant le bit de parité). En revanche, lors de la transmission de cette lettre Q, le bit 4 a été inversé de 1 à 0, alors que le bit de parité est resté à l'état 1. Lorsque la parité de cet octet altéré est vérifiée, le programme ou le matériel découvrira que cet octet a été altéré et il est rejeté. En réfléchissant à la question, vous noterez également que, même si le bit de parité est inversé lors d'une transmission, le fait qu'une erreur soit survenue serait encore détecté par le processus de vérification de parité, et l'octet serait rejeté.

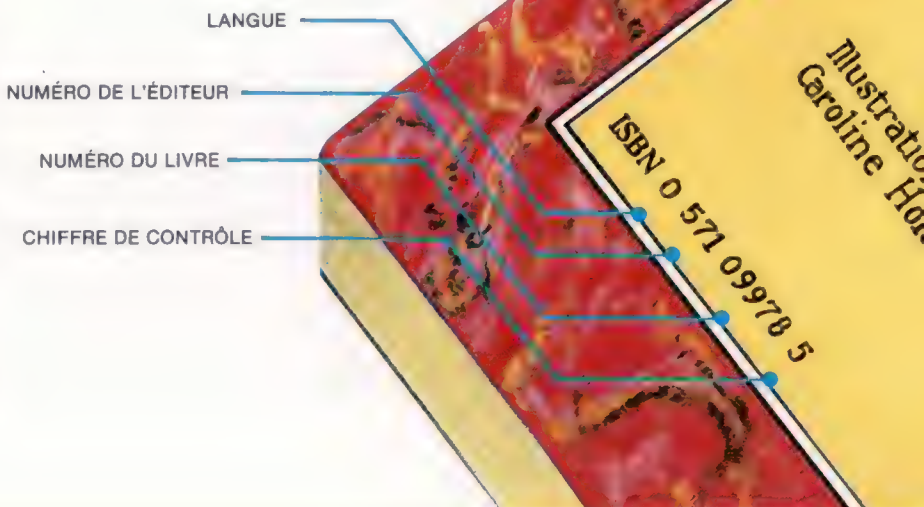
Si vous examinez les codes ASCII utilisés dans votre propre ordinateur, vous découvrirez probablement que le bit 7 (le bit le plus significatif, ou « de poids fort ») est en fait utilisé, mais non comme bit de parité. Cela a pour objet de permettre à l'ordinateur de posséder un jeu de caractères additionnel (généralement un jeu de caractères graphiques), et parce que les erreurs de transmission à l'intérieur même de l'ordinateur sont très rares. La vérification de parité est normalement utilisée uniquement pour les transmissions sur de longues distances, ou lorsque les données sont enregistrées sur des surfaces magnétiques (comme les cassettes et les disquettes) où des erreurs de bits sont également possibles.

La vérification de la parité est parfaite pour indiquer qu'un octet a mal été transmis, mais elle n'indique pas quel bit de l'octet a mal été transmis. L'ordinateur récepteur ne peut donc pas corriger l'erreur. Pis encore, si 2 bits d'un octet ont été inversés, cette erreur de transmission ne pourra pas être repérée.

Mais, dans les cas où le dispositif récepteur détecte une erreur, il peut envoyer un message d'erreur et le logiciel peut retransmettre l'octet altéré.

Simple vérification

Le dernier chiffre du numéro ISBN (International Standard Book Number) d'un livre équivaut au bit de parité d'un ordinateur. Multipliez le premier chiffre (0 ici) par 10, le second (5) par 9, et ainsi de suite, puis additionnez ces résultats. Vous découvrirez que le chiffre de contrôle porte le total à un multiple de 11.



Fichier fractionné

Nous continuons notre projet de programmation d'un carnet d'adresses informatisé en examinant comment notre fichier de données doit être fractionné en enregistrements et en champs.

Le dernier cours de programmation en BASIC s'est terminé en expliquant comment perfectionner les éléments de notre exercice de programmation en passant par plusieurs niveaux de pseudo-langage, jusqu'au point où les formulations ainsi obtenues pouvaient être traduites en BASIC. Nous allons d'abord réviser cet exercice, puis donner ensuite des solutions possibles. Le premier « Énoncé des objectifs » de cet exercice était :

ENTRÉE

Un nom (tout format)

SORTIE

1. Un prénom
2. Un nom

Dans le premier niveau de perfectionnement, nous avons fractionné cela en six étapes (nous avons découvert plus tard que la sixième étape pouvait être omise). Voici ces étapes :

1. Lire le nom (* LIRE *)
2. Convertir en majuscules (* CONVERSION *)
3. Trouver le dernier espace (* ESPACE *)
4. Lire le nom de famille (* LIRE NOM DE FAMILLE *)
5. Lire le prénom (* LIRE PRÉNOM *)
6. Rejeter les caractères non alphabétiques du prénom

Toutes ces interventions seront traitées comme des sous-programmes, et le nom que nous attribuons à chaque sous-programme est donné entre parenthèses. Malheureusement, la plupart des versions de BASIC ne sont pas en mesure d'appeler des sous-programmes en utilisant leur nom. Il sera donc nécessaire d'insérer des numéros de ligne à la suite des instructions GOSUB respectives lors de l'écriture finale du programme. Il est cependant beaucoup plus facile, pendant le développement du programme, de désigner les sous-programmes par leur nom. Ces noms pourront plus tard être mentionnés dans des instructions REM. Les noms des sous-programmes sont toujours précédés et suivis d'un astérisque. Dans des langages qui peuvent appeler des sous-programmes à l'aide de leur noms (comme le PASCAL), on nomme généralement ce genre de sous-programme une « procédure ».

Même si votre BASIC n'est pas à même de manipuler des procédures, il est préférable que vous simuliez cette possibilité lors du stade de programmation en pseudo-langage. De façon similaire, il est possible que votre version de BASIC ne soit pas en mesure de manipuler des longs noms de variables comme COMPTE ou NOMRUE\$, mais au niveau du pseudo-langage il est préférable de faire comme si cela était possi-

ble. Essayez de choisir des noms suffisamment descriptifs. Il est beaucoup plus clair de nommer une variable temporaire de chaîne TEMPCHAINE\$ que de la nommer XV\$.

DEUXIÈME PERFECTIONNEMENT

3. (Trouver le dernier espace)

DÉBUT

BOUCLE quand des caractères non lus restent dans NOM\$

SI caractère = « »

ALORS noter la position dans la variable

SINON ne rien faire

FINSI

FINBOUCLE

FIN

TROISIÈME PERFECTIONNEMENT

3. (Trouver le dernier espace)

DÉBUT

LIRE NOM\$

BOUCLE quand des caractères non lus restent dans NOM\$

POUR L = 1 à longueur de NOM\$

LIRE caractère de NOM\$

SI caractère = « »

ALORS COMPTE = position du caractère

SINON ne rien faire

FINSI

FINBOUCLE

FIN

Nous pouvons maintenant passer du pseudo-langage au langage de programmation :

```
10 INPUT « ENTREZ LE NOM COMPLET »; NOM$
```

```
20 FOR L = 1 TO LEN (NOM$)
```

```
30 LET CAR$ = MID$(NOM$, L, 1)
```

```
40 IF CAR$ = « » THEN LET COMPTE = L
```

```
50 NEXT L
```

```
60 PRINT « DERNIER ESPACE EN POSITION »; COMPTE
```

```
70 END
```

Notez que la ligne 10 est une entrée factice pour tester la routine; la ligne 60 est également une sortie factice à des fins de vérification; la ligne 70 devra devenir une instruction RETURN lorsque cette routine sera utilisée comme sous-programme. Essayons maintenant la même chose pour l'étape 4 :

DEUXIÈME PERFECTIONNEMENT

4. (Lire le nom de famille)

DÉBUT

Affecter les caractères à la droite du dernier espace à FNOM\$

END

TROISIÈME PERFECTIONNEMENT

4. (Lire le nom de famille)

DÉBUT

LIRE NOM\$



Localiser le dernier espace (appeler * ESPACE * sous-programme)
 BOUCLE quand des caractères non lus restent après le dernier espace
 LIRE les caractères et ajouter à FNOM\$
 FINBOUCLE
 FIN

Avant de traduire ceci en BASIC, vous devriez noter certains pièges éventuels. Pour localiser le dernier espace dans la dernière étape de perfectionnement ci-dessus, le pseudo-langage appelle le sous-programme * ESPACE *, mais il ne serait pas possible d'écrire ceci en BASIC et de le tester si le sous-programme * ESPACE * n'avait pas été écrit. Comme règle générale, il est préférable de ne pas coder les modules en BASIC (ou en tout autre langage évolué) tant que le programme n'a pas été développé au complet dans le pseudo-langage. Si cependant, vous désirez tester un module, il est possible que vous ayez à écrire certaines valeurs de variables factices ainsi que des instructions d'entrée et de sortie factices. Dans l'exemple précédent, COMPTE est la variable qui renferme la valeur de la position du dernier espace dans NOM\$. A des fins de vérification, nous pourrions tricher un peu en supposant que la routine qui se charge de définir cette valeur fonctionne adéquatement :

```
10 LET NOM$ = JEAN DUPONT
20 LET COMPTE = 4
30 FOR L = COMPTE + 1 TO LEN (NOM$)
40 LET FNOM$ = FNOM$ + MID$ (NOM$, L, 1)
50 NEXT L
60 PRINT « LE NOM DE FAMILLE EST »; FNOM$
70 END
```

Voici comment trouver le prénom (étape 5). Rappelons que nous avons défini le prénom comme étant l'enchaînement de tous les caractères alphabétiques apparaissant avant le dernier espace du nom complet. Tout caractère non alphabétique doit être rejeté.

DEUXIÈME PERFECTIONNEMENT

5. (Lire le prénom)
 DÉBUT
 BOUCLE quand des caractères non lus restent dans NOM\$ jusqu'au dernier espace
 Analyser les caractères
 SI le caractère est non alphabétique
 ALORS ne rien faire
 SINON ajouter le caractère à PRÉNOM\$
 FINSI
 FINBOUCLE
 FIN

TROISIÈME PERFECTIONNEMENT

5. (Lire le prénom)
 DÉBUT
 BOUCLE tant qu'il reste des caractères jusqu'à COMPTE
 TEMPCAR\$ = L^{ème} caractère de la chaîne
 SI TEMPCAR\$ n'est pas une lettre
 ALORS ne rien faire
 SINON PRÉNOM\$ = PRÉNOM\$ + TEMPCAR\$
 FINSI
 FINBOUCLE
 FIN

Nous sommes maintenant prêts à traduire ceci en BASIC, mais, comme étape intermédiaire, nous utiliserons un format structuré d'instructions BASIC non numérotées afin de comparer le programme avec la structure précédente :

CODAGE

```
5. (Lire le prénom)
REM DÉBUT
REM BOUCLE
FOR L = 1 TO COMPTE - 1
  LET TEMPCAR$ = MID$ (NOM$, L, 1)
  LET CAR = ASC (TEMPCAR$)
  SI CAR > 64 THEN PRÉNOM$ = PRÉNOM$
  + CHR$ (CAR)
  REM FINSI
NEXT L : REM FINBOUCLE
REM END
```

En BASIC ordinaire cela serait :

```
10 FOR L = 1 TO COMPTE - 1
20 LET TEMPCAR$ = MID$ (NOM$, L, 1)
30 LET CAR = ASC (TEMPCAR$)
40 SI CAR > 64 THEN PRÉNOM$ = PRÉNOM$
  + CHR$ (CAR)
50 NEXT L
60 END
```

Tel quel, cependant, ce programme ne fonctionnerait pas. Il y a trois problèmes : une valeur doit être affectée à COMPTE; il n'y a aucune instruction d'entrée de nom (affectation d'une chaîne à NOM\$); et il n'y a aucune instruction de sortie qui puisse nous permettre de vérifier le bon fonctionnement du programme.

Si cette routine faisait partie d'un sous-programme, les paramètres qui lui seraient transmis (l'entrée) et les paramètres qui sont transmis par le sous-programme (la sortie) devraient être manipulés ailleurs dans le programme. Ceci est très important : la circulation de l'information à l'intérieur d'un programme doit être planifiée avec soin avant de commencer à transcrire le programme en BASIC. Cela est particulièrement vrai lorsque nous utilisons des variables (COMPTE par exemple) et que le même nom de variable est utilisé dans différentes parties du programme. Cela ne sert à rien d'appeler un sous-programme qui utilise une variable comme COMPTE si le sous-programme n'a aucun moyen de savoir quelle doit être la valeur de cette variable. Si un sous-programme initialise la valeur de COMPTE, cette valeur demeurera la même jusqu'à ce qu'une nouvelle valeur soit affectée, peut-être dans un autre sous-programme. Voilà pourquoi il n'est pas recommandé de sortir au milieu d'une boucle puisque la valeur de la variable de la boucle est inconnue à ce moment. Examinez les conséquences qu'entraîneraient ces deux fragments de programme situés à l'intérieur de deux sous-programmes différents :

Partie du sous-programme X

```
FOR L = 1 TO LEN (MOT$)
LET CAR$ = MID (MOT$, L, 1)
SI CAR$ = « . » THEN GOTO 1550
NEXT L
```



Partie du sous-programme Y

```
FOR Q = 1 TO LIMIT
LET A (Q) = P (Q)
NEXT Q
```

Cette partie du sous-programme Y lit des valeurs dans un tableau indicé, où l'indice est défini par la variable L. Si le sous-programme Y est appelé après le sous-programme X, et si la condition de test du sous-programme X a été satisfaite (un caractère est un « . »), la valeur de L est parfaitement imprévisible et nous ne pouvons absolument pas savoir quelles valeurs du tableau sont affectées dans le sous-programme Y. En plus de cette erreur de branchement hors d'une boucle, ce sous-programme utilise aussi une instruction GOTO, et cette pratique devrait être évitée. L'instruction GOTO peut entraîner diverses confusions, et devrait être évitée lorsque cela est possible.

Pour éviter toute confusion lors de l'utilisation de variables, il est préférable de faire une liste de variables pendant les étapes de pseudolangage du développement du programme, et de mentionner dans cette liste la fonction de chaque variable. Certains langages (mais pas le BASIC) permettent de déclarer des variables comme étant « locales » ou « globales », c'est-à-dire que leurs valeurs peuvent s'appliquer uniquement dans une partie du programme (locales) ou dans l'ensemble du programme (globales). De nombreuses variables, comme celles utilisées dans des boucles (par exemple, le L dans LET L = 1 TO 10), sont presque toujours locales. Il est donc souvent indiqué d'initialiser la valeur de la variable avant qu'elle ne soit utilisée (par exemple, LET L = 0). Certains langages, comme le PASCAL, insistent sur ce point, et bien que le BASIC suppose toujours que la valeur initiale d'une variable soit zéro (sauf si elle a déjà été définie), l'initialisation est toujours recommandée.

Jusqu'ici, nous avons formulé une définition d'un nom acceptable pour les besoins de notre carnet informatisé, et créé les routines permettant d'effectuer les diverses manipulations requises par notre programme. Maintenant, prenons de nouveau du recul par rapport aux détails du programme codé, et examinons la structure des enregistrements du fichier de notre carnet d'adresses.

A l'intérieur de chaque enregistrement nous avons des champs. Un champ est une ou plusieurs lignes d'informations associées à l'intérieur d'un enregistrement. Chacun des enregistrements de notre fichier ADRESSE comprendra les champs suivants : NOM, ADRESSE et TÉLÉPHONE. Voici un enregistrement typique :

```
Henri Faria
1, rue Gustave-Flaubert
78300 Poissy
(3) 916.58.69
```

Dans cet enregistrement, il y a trois champs : le champ du nom, qui comprend des lettres alphabétiques (et possiblement des apostrophes pour

des noms comme O'Toole); le champ de l'adresse, qui comprend quelques nombres et de nombreuses lettres; et le champ du numéro de téléphone, qui ne comprend que des nombres (ignorons le problème posé par la présence de point et de parenthèses comme dans (3) 916.58.69). Avant de commencer à écrire un programme qui puisse manipuler avec souplesse des informations complexes comme celles-ci, nous devons décider comment représenter les données à l'intérieur de l'ordinateur. Il serait possible de faire en sorte que toute l'information d'un enregistrement ne soit contenue que dans une seule chaîne de caractères. Mais cette approche rendrait extrêmement difficile l'extraction d'une information spécifique.

```
PERCIVAL R. BURTON
1056 AVENUE OF THE AMERICA
RIO DEL MONTENEGRO
CALIFORNIA
U.S.A.
(415) 884 5100
```

Si nous effectuons une recherche dans les enregistrements pour trouver le numéro de téléphone de Monsieur Burton, serait-il prudent de supposer que les 14 derniers caractères de l'enregistrement représentent le numéro de téléphone? Et si nous avons inclut l'indicatif international, comme ceci : 19.1 (415) 884 5100? Le numéro compterait alors 18 caractères. Pour éviter ce problème, le numéro de téléphone est placé dans un champ séparé, et le programme nous communiquera tous les caractères (ou nombres) de ce champ lorsque nous le demanderons.

Cette approche pose le problème suivant : nous devons trouver un moyen d'établir une relation entre les divers champs, afin qu'en mentionnant un champ (le champ du nom par exemple) nous puissions obtenir les autres champs de l'enregistrement. Une manière d'y parvenir serait de créer un autre champ associé à l'enregistrement et qui ne remplirait qu'une fonction d'indexation.

Lucien Feugière	Champ NOM
15, place de la République	Champ RUE
31000	Champ CODE POSTAL
TOULOUSE	Champ VILLE
(61) 45.28.65	Champ TÉLÉPHONE
015	Champ INDEX

Si nous connaissons le nom de cette personne et désirions connaître son numéro de téléphone, nous n'aurions qu'à chercher, dans le tableau renfermant les noms, un nom qui correspond à celui recherché. Nous devrions alors trouver dans quel élément du tableau NOM nous avons trouvé le nom recherché — ici, le nombre 15. Puis nous n'aurions plus qu'à consulter le quinzième élément du tableau TÉLÉPHONE pour obtenir le numéro de téléphone recherché.

Si nous avons de nombreux amis habitant à Rambouillet, nous pourrions demander au programme de rechercher chaque apparition de



« Rambouillet » dans le champ VILLE et de noter l'emplacement de chaque apparition de Rambouillet. Tout ce que le programme aurait alors à faire pour imprimer les noms et adresses de tous ces amis serait d'extraire tous les éléments ayant le même numéro de tableau que chaque enregistrement « Rambouillet » découvert. Avec cette approche, il ne serait plus nécessaire d'inspecter le champ INDEX. En plus, cette technique a le mérite d'être assez simple.

Exercices

Supposons que des enregistrements comportant les champs suivants seraient adéquats pour notre carnet d'adresses informatisé :

- Champ NOM
- Champ RUE
- Champ CODE POSTAL
- Champ VILLE
- Champ TÉLÉPHONE

Supposons que l'une des options offertes par un menu du carnet d'adresses informatisé est :

5. Créer un nouvel enregistrement

Vous tapez 5 et le programme effectue un bran-

chement dans la partie du programme qui crée de nouveaux enregistrements (vous pouvez supposer que le carnet d'adresses est encore vide). Puisque le programme est entièrement piloté par menus, il sollicitera toujours l'entrée attendue avec des messages comme ceux-ci : ENTREZ LE NOM, ENTREZ LA RUE, et ainsi de suite. Voici une liste des résultats escomptés :

1. Un élément dans un tableau pour le nom
2. Un élément dans un tableau pour la rue
3. Un élément dans un tableau pour le code postal
4. Un élément dans un tableau pour la ville
5. Un élément dans un tableau pour le numéro de téléphone

Votre tâche est de résoudre ce problème au moyen d'un processus de programmation descendante en utilisant un pseudo-langage, jusqu'à ce que vous soyez en mesure de convertir directement en BASIC la formulation obtenue. Vous pouvez définir les règles de votre pseudo-langage ; nous vous suggérons cependant d'utiliser des lettres majuscules pour les mots clés comme SI, BOUCLE et ainsi de suite, et des lettres minuscules pour décrire en français ordinaire les opérations à effectuer.

Variantes de basic



Étape 3

```
10 INPUT « ENTREZ NOM COMPLET »; N$
15 LET COMPTE = 0
20 FOR L = 1 TO LEN N$
30 LET C$ = FN M$(N$, L, 1)
40 IF C$ = ' ' THEN LET COMPTE = L
50 NEXT L
60 PRINT « LE DERNIER ESPACE EST
  EN POSITION »; COMPTE
70 STOP
9990 DEF FN M$(X$,P,N) = X$(P TO P+N-1)
9991 DEF FN L$ = X$(TO N)
9992 DEF FN R$ = X$(LEN X$-N+1 TO)
```

Dans ce projet de programmation, les fonctions de chaîne MID\$, LEFT\$ et RIGHT\$ seront souvent utilisées, il est donc préférable de les créer pour le Spectrum à l'aide de l'instruction DEF FN dans les lignes 9990 à 9992. Dès que ces lignes font partie de votre programme, vous pouvez remplacer MID\$ par FN M\$, LEFT\$ par FN L\$, et RIGHT\$ par FN R\$; les lignes de ces instructions n'ont pas à être nécessairement numérotées de 9990 à 9992, mais elles doivent être insérées si vous désirez utiliser ces fonctions.

Étape 4

```
5 LET S$ = « »
10 LET N$ = « JEAN DUPONT »
20 LET COMPTE = 4
30 FOR L = COMPTE + 1 TO LEN N$
40 LET F$ = F$ + FN M$(N$,L,1)
50 NEXT L
60 PRINT « LE NOM DE FAMILLE »; F$
70 STOP
```

Étape 5

```
5 LET P$ = « »
10 FOR L = 1 TO COMPTE - 1
20 LET T$ = FN M$(N$,L,1)
```

```
30 LET CAR = CODE T$
40 IF CAR > 64 THEN LET P$ = P$ + CHR$ CAR
50 NEXT L
60 STOP
```

Dans ce fragment, puisque le Spectrum ne peut employer que des variables de chaîne à une lettre, F\$ remplace FNOM\$ sur le Spectrum, et P\$ remplace PRÉNOM\$.

Partie du sous-programme X

```
FOR L = 1 TO LEN W$
LET C$ = FN M$(W$,L, 1)
IF C$ = « . » THEN GOTO 1550
NEXT L
```

Partie du sous-programme Y

```
FOR Q = 1 TO LIMIT
LET A (Q) = P(Q)
NEXT Q
```

Le BBC est l'un des seuls ordinateurs domestiques qui puissent prendre en charge les noms de variables longs comme PRÉNOM\$. Le Spectrum accepte les noms longs de variables numériques, mais uniquement des variables de chaîne à 1 caractère. Le Dragon-32, le Vic-20 et le Commodore 64 acceptent des noms de variable longs, mais seuls les deux premiers caractères sont significatifs, PRÉNOM\$ est donc accepté mais désigne la même variable que PRÉTS, puisque ces deux noms de variable commencent par PR. Sur l'Oric-1, les noms de variables ne peuvent compter plus de deux caractères (une lettre puis un chiffre ou une lettre), alors que le LYNX n'accepte que des noms de variables à une seule lettre. Malheureusement, le ZX81 n'offre ni les fonctions de manipulation de chaîne standard ni la fonction DEF/ NF. Nous ne mentionnerons donc pas cette machine dans nos encadrés.



D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z



Papier-calque

Il est possible de transférer des images dans votre ordinateur au moyen d'un numériseur, ou tablette graphique.

Les fonctions graphiques font partie des attributs les plus importants de la génération actuelle d'ordinateurs. A l'aide de quelques commandes très simples, il est possible de créer des motifs ou des dessins dont les couleurs peuvent être modifiées à volonté. Tout cela requiert une certaine connaissance de la programmation, puisqu'il n'est pas encore possible de créer

Curseur
Cet appareil est déplacé par la main au-dessus de l'image qui va être numérisée.

d'abord une image sur papier et de la charger dans l'ordinateur lorsque le travail est terminé. Les crayons optiques facilitent l'édition et la manipulation d'une image à l'écran, mais ils ne peuvent pas être utilisés pour copier une image tracée sur une feuille de papier.

Les concepteurs de voitures, d'avions et de microprocesseurs, ainsi que les architectes d'intérieur, les architectes paysagistes et les dessinateurs de mode peuvent tous exploiter le potentiel graphique de l'ordinateur. Dès que le dessin est stocké dans la mémoire de l'ordinateur, diverses modifications peuvent être tentées sans que cela n'implique de gaspillage de matières premières. Il est donc nécessaire de disposer d'un périphérique d'entrée capable de traduire les lignes et les courbes du dessin en un langage que l'ordinateur peut comprendre.

Dans le secteur professionnel, la « tablette graphique » a fait son apparition peu de temps après l'ordinateur. Cependant, ce n'est que très récemment que des solutions économiques ont été offertes à l'utilisateur d'ordinateur domestique. Les tablettes graphiques à haute précision, aussi nommées « numériseurs » parce qu'elles convertissent des formes analogiques et des images en une information numérique, font appel à une vaste gamme technique pour produire l'information requise. Les systèmes les plus précis peuvent obtenir une résolution atteignant un quart de millimètre, ce qui est suffisant pour les ingénieurs et les dessinateurs. Tous les numériseurs comportent une tablette plate sur laquelle est posé un dessin. Une pointe de lecture, soit un stylo ordinaire ou un disposi-

Mire
Une mire et une loupe qui aide à positionner le curseur plus précisément. Les résolutions allant jusqu'à 0,25 mm sont chose courante.

Boutons d'entrée de données
La plupart des curseurs possèdent plus d'un bouton-poussoir; ces boutons permettent à l'opérateur d'indiquer qu'un point particulier doit être enregistré. Dans un autre mode, le numériseur effectue une lecture continue. (Cl. David Wecks.)

Bobine d'émission
Un signal à haute fréquence est produit par cette bobine et capté par le réseau.

tif électronique sophistiqué, trace alors les contours de l'image. La position de la pointe est détectée par le numériseur et transmise à l'ordinateur sous la forme de coordonnées.

Les deux systèmes les plus précis — magnétique et capacitif — possèdent un réseau de fils encastré dans la tablette. Dans le système magnétique, la pointe de lecture consiste en une petite loupe munie d'une mire avec laquelle le contour de l'image est tracé. Une bobine entoure cette loupe et émet un signal de faible amplitude et de haute fréquence. Le signal est détecté par le réseau de la tablette, ce qui permet ainsi de mesurer la position exacte de la pointe de lecture. Le système capacitif fonctionne de façon inverse : une série d'impulsions codées sont transmises dans un réseau de fils et ce signal est capté par la pointe de lecture.

Le système acoustique offre une autre méthode. La pointe de lecture est chargée électrostatiquement, et est ensuite appliquée contre

Interface
Les numériseurs sont généralement connectés à un ordinateur à l'aide d'un port série ou parallèle standard.



Tablette

L'image devant être numérisée est posée à plat sur la tablette, une charge électrostatique est appliquée à la tablette afin de plaquer la feuille sur la surface du numériseur.

la tablette, ce qui produit une étincelle minuscule. L'onde acoustique créée par l'étincelle est captée par deux microphones : les décalages réciproques de réception permettent de calculer la position exacte de la pointe. Cela offre, entre autres choses, la possibilité de numériser la troisième dimension, au moyen d'un signal traversant l'objet.

Parmi les produits de bas de gamme, on retrouve la tablette sensible à la pression : l'image est posée sur la tablette et la pointe en suit les contours. L'opérateur doit appliquer



Lecture de carte

L'une des utilisations les plus répandues est de recueillir des données sur des cartes et des plans. Ici l'ordinateur est utilisé pour prédire l'emplacement d'un nouveau champ pétrolier. (Cl. Simon Lewis.)

Carte de traitement

Cette carte de circuits imprimés renferme un microprocesseur, une ROM et une RAM. Ces composants permettent au numériseur d'envoyer l'information en coordonnées X-Y.

de l'ordre de 5 %. Des pantographes sophistiqués, dont la conception est fondée sur une lecture optique de la position des articulations, peuvent offrir de bien meilleurs résultats.

Les tablettes optiques comportent un réseau de faisceaux infrarouges qui détectent la position de la pointe. Ils ne sont pas aussi sensibles que les autres systèmes, mais ils permettent de désigner un élément d'un programme à l'aide d'un doigt. Dans certaines applications, les sources et capteurs infrarouges sont placés autour de la bordure de l'écran, créant ainsi un véritable écran interactif où des images peuvent être tracées simplement en déplaçant le doigt sur l'écran.

Les données brutes produites par une tablette graphique doivent être traduites en une information pouvant être affichée. C'est pourquoi la plupart des produits commerciaux sont livrés avec le logiciel nécessaire. Les tablettes graphiques ne servent pas uniquement à entrer des données. Dès que les données sont en mémoire, la tablette peut être utilisée comme un outil d'édition, permettant de modifier les couleurs et les formes du dessin. La surface de la tablette peut être programmée pour servir de menu où sont sélectionnées les options standard du programme; le clavier ne sert plus alors qu'à sélectionner les fonctions principales. Les systèmes d'animation par ordinateur possèdent tous une tablette graphique de haute qualité comme principal périphérique d'entrée.

une plus grande pression sur le papier que dans les autres systèmes. Deux feuilles électriquement conductrices sont séparées par un isolant cellulaire et deux signaux différents à haute fréquence sont transmis dans les feuilles. Le signal détecté par la pointe lorsqu'elle établit une connexion électrique entre les deux feuilles permet de mesurer sa position.

Les numériseurs les plus simples et les moins chers sont les pantographes — dont la conception s'inspire des anciens instruments à dessin —, composés de tiges articulées. Ils utilisent la géométrie analytique pour calculer directement la position de la pointe. Des résistances variables montées sur les deux articulations permettent d'obtenir des tensions proportionnelles aux angles de l'« épaule » et du « coude » du bras articulé. La résolution du pantographe est fonction de la précision des résistances et des articulations mécaniques; elle est généralement

Réseau de réception

Encastré dans la tablette, un réseau de fils capte le signal émis par la bobine. L'espacement des fils du réseau est beaucoup plus grossier que la résolution du numériseur parce que le circuit de traitement peut interpoler des valeurs à partir de la force relative du signal capté par deux fils adjacents.



Gottfried Leibniz

1646
Né le 1^{er} juillet à Leipzig.

1661
S'inscrit à l'université de Leipzig et obtient un diplôme à l'âge de dix-sept ans.

1660-1670
Avocat et diplomate. Il publie un ouvrage *De arte combinatoria*.

1672
Il développe à Paris le principe de la raison suffisante.

1673
Il présente une calculatrice à la Société royale d'Angleterre.

1675
Découvre le calcul différentiel et intégral indépendamment de Newton.

1676
Étudie la dynamique à travers les notions d'énergie cinétique.

1678
Est nommé bibliothécaire et conseiller du duc de Hanovre.

1679
Développe les mathématiques binaires.

1683
Publie un pamphlet où il attaque Louis XIV.

1690-1700
Termine la généalogie de la maison de Hanovre et entreprend une étude historique du monde. Commence à s'intéresser à la linguistique et à l'origine des langues.

1700
Fonde l'Académie des sciences de Berlin.

1714
Établit le droit de succession de Georges 1^{er} au trône d'Angleterre, laissé vacant par la mort de la reine Anne.

1716
Meurt à Hanovre, le 14 novembre. (Cl. Musée des sciences).



Les travaux de ce penseur du XVII^e siècle aident les recherches sur l'ordinateur de la cinquième génération.

Gottfried Wilhelm Leibniz, un des plus grands esprits scientifiques de son époque, est né à Leipzig en 1646 et mort à Hanovre en 1716. Pendant ses soixante-dix ans d'existence, il inventa le calcul différentiel et intégral, travailla sur la dynamique et apporta sa contribution à la géologie, à la théologie, à l'histoire, à la linguistique et à la philosophie. Mais ce que nous devons surtout mentionner, c'est qu'il introduisit des notions qui allaient être déterminantes dans la création de l'ordinateur.

Leibniz commença ses voyages à l'âge de vingt ans, après que l'université de Leipzig refusa de lui octroyer un doctorat de droit en raison de son jeune âge. Tout au long de sa vie, n'ayant trouvé personne pour financer ses travaux, il fut obligé d'accepter des emplois, ce qui gêna sa recherche scientifique. Au début de la trentaine, il fut avocat et diplomate, puis plus tard il devint bibliothécaire et conseiller royal.

Sa culture universelle l'entraîna dans de nombreux voyages en Europe et lui fit rencontrer

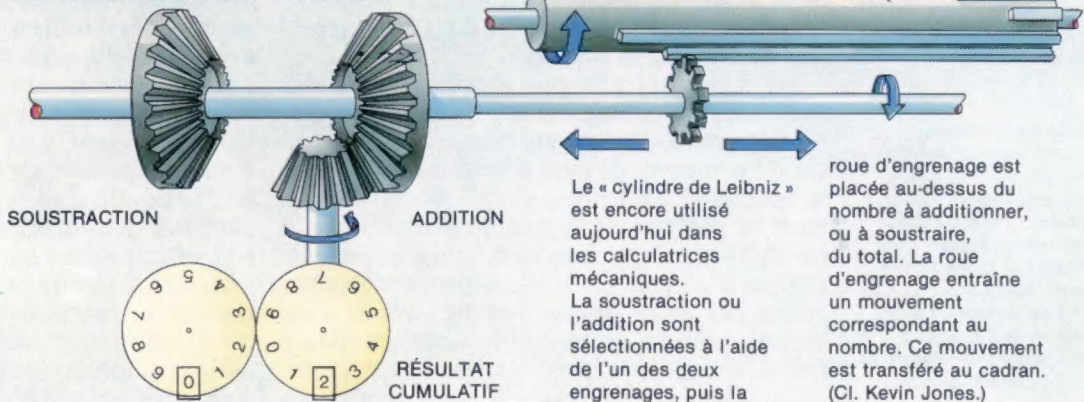
tous les grands penseurs de son époque. Il fut également un écrivain prolifique et entretenit une correspondance avec plus de six cents personnes. Sa contribution la plus importante à la philosophie est intervenue en 1672 lorsqu'il formula le principe de la raison suffisante. Exposé très simplement, ce principe attribue une raison pour chaque chose et justifie la nature de chaque élément composant un univers équilibré.

Leibniz s'intéressa un certain temps aux mathématiques et essaya de perfectionner la pascaline, la machine à additionner inventée par Blaise Pascal en 1642. Il chercha à accroître le potentiel de ce mécanisme afin de lui permettre d'effectuer des multiplications et des divisions. Il y parvint en concevant un dispositif mécanique nommé « cylindre de Leibniz » (voir ci-dessous). Cette machine représentait une réelle percée scientifique à cette époque. Précédemment, en raison de la complexité de manipulation des chiffres romains, la multiplication n'était enseignée que dans les établissements de haut savoir. L'arrivée de cette machine pouvant multiplier rendit l'arithmétique accessible à un plus grand nombre de personnes; lorsque Leibniz en eut terminé la mise au point, il se détourna de la base dix et commença à examiner et à formuler les mathématiques binaires.

La plus grande ambition de Leibniz était de concevoir un langage universel qui intégrerait la précision et la clarté des mathématiques, et qui permettrait de résoudre tous les problèmes auxquels l'homme pouvait être confronté. Pour ce faire, son langage allait devoir utiliser des symboles abstraits représentant les « atomes » de l'intelligence, et manipuler ces symboles en obéissant à un ensemble de règles. Il échoua dans cette tentative; mais ses idées furent reprises plus modestement au début du XX^e siècle par Bertrand Russell, qui essaya d'expliquer les mathématiques au moyen d'un langage logique.

Au cours de ces dernières années, les scientifiques impliqués dans les projets à long terme visant à créer les ordinateurs de la cinquième génération ont manifesté un regain d'intérêt pour les travaux de Leibniz.

La calculatrice de Leibniz



abc

INFORMATIQUE

Pour classer et présenter dans votre bibliothèque les fascicules de votre collection, des reliures mobiles pratiques et élégantes sont en vente chez tous les marchands de journaux.

Dans chaque reliure, vous trouverez, dans l'enveloppe qui contient les deux lames métalliques de la reliure, un décalque portant les numéros 1 à 8, qui vous permettra de marquer vous-même le dos de chaque volume. Pour relier les 12 fascicules qui composent un volume, vous devrez en retirer les couvertures sans endommager les agrafes métalliques. Le numérotage de la reliure puis la mise en place des fascicules doivent être effectués selon les instructions ci-dessous.

1

Disposez à plat la reliure. Enlevez le papier de protection du décalque. Positionnez le décalque en faisant coïncider l'écran qui entoure le numéro choisi avec l'écran situé sur le dos de la reliure.

2

Avec la pointe d'un stylo à bille, frottez régulièrement le numéro à transférer, en exerçant une certaine pression et en débordant légèrement.

3

Enlevez doucement le support : le numéro est reporté sur la reliure. Posez dessus le papier de protection du décalque et frottez largement avec un objet poli ou arrondi, de manière à assurer la parfaite adhérence des caractères transférés.

4

Retournez la reliure mobile. Introduisez, d'un côté seulement, dans les encoches pratiquées dans l'épaisseur de la couverture, l'une des extrémités des deux lames d'acier livrées avec la reliure.

5

Vérifiez le bon ordre des fascicules, puis passez les deux lames d'acier dans les agrafes supérieures et inférieures.

6

Introduisez les extrémités libres des lames d'acier dans les encoches correspondantes de la reliure.

Ce système original, sans mécanisme visible, donne au volume l'apparence d'une reliure classique. Nous vous conseillons d'avoir toujours une reliure d'avance pour mieux protéger vos fascicules au fur et à mesure de leur parution.

Protégés par une élégante reliure, vos numéros d'aBc Informatique seront plus faciles à consulter

Pour classer, répertorier, protéger vos fascicules d'ABC Informatique, les Editions Atlas vous proposent des reliures élégantes, sobres, qui s'insèrent parfaitement dans votre bibliothèque. Chacune contient 12 fascicules,

les maintient, les préserve. Un système simple, résistant, vous permet de les assembler facilement. Elles sont en vente en permanence chez votre marchand de journaux. Demandez-les !



Chaque
reliure:
40 FF
295 FB
18 FS

Cod. N.M.P.P. :
6103

EDITIONS
ATLAS